Discriminatively-guided Deliberative Perception for Pose Estimation of Multiple 3D Object Instances

Venkatraman Narayanan and Maxim Likhachev
The Robotics Institute, Carnegie Mellon University
{venkatraman,maxim}@cs.cmu.edu

***Abstract*—We introduce a novel paradigm for model-based multi-object recognition and 3 DoF pose estimation from 3D sensor data that integrates exhaustive global reasoning with discriminatively-trained algorithms in a principled fashion. Typical approaches for this task are based on scene-to-model feature matching or regression by statistical learners trained on a large database of annotated scenes. These approaches are fast but sensitive to occlusions, features, and/or training data. Generative approaches, on the other hand, e.g., methods based on rendering and verification, are robust to occlusions and require no training, but are slow at test time. We conjecture that robust and efficient perception can be achieved through a combination of generative methods and discriminatively-trained approaches. To this end, we introduce the Discriminatively-guided Deliberative Perception (D2P) paradigm that has the following desirable properties: a) D2P is a *single* search algorithm that looks for the 'best' rendering of the scene that matches the input, b) can be guided by *any* and *multiple* discriminative algorithms, and c) generates a solution that is provably bounded suboptimal with respect to the chosen cost function. In addition, we introduce the notions of *completeness* and *resolution completeness* for multi-object pose estimation problems, and show that D2P is resolution complete. We conduct extensive evaluations on a benchmark dataset to study various aspects of D2P in relation to existing approaches.**

I. Introduction

The predominant objective of robotic perception systems is to identify and locate quantities of interest in the physical world. In many cases, these quantities of interest are specific object instances whose 3D models are available, such as objects on a warehouse shelf, or parts on an assembly line. In the early days of machine vision [21], researchers abstracted the instance detection problem as a "blocks world" problem in which the task was to identify the pose of specific blocks in a 2D image. The methodology used was one of "inverse optics" or analysis-by-synthesis, where a search was performed for a configuration of blocks that best explained the input 2D image. The definition of "best explanation" was chosen to be that of a matching between model and scene primitives such as edges and corners, so that effects of illumination on the raw pixel intensities could be bypassed.

While the methodology of the blocks-world research seems appropriate for instance-detection problems in robotics, it has not found widespread adoption for two major reasons: i) the problem of determining primitives/features that are invariant to nuisance parameters has been hard enough for real-world images that it has become its own line of research, and ii) performing an exhaustive search for the best global hypothesis is computationally infeasible for real-world systems. Consequently, much of the recent work in 3D object detection and pose estimation has dealt with developing better feature representations [25, 2, 4], or learning algorithms that can generalize from limited training data [26, 8]. This is also evident from the instance-detection algorithms used in the recent Amazon Picking Challenge [7], most of which were based on scene-to-model feature matching.

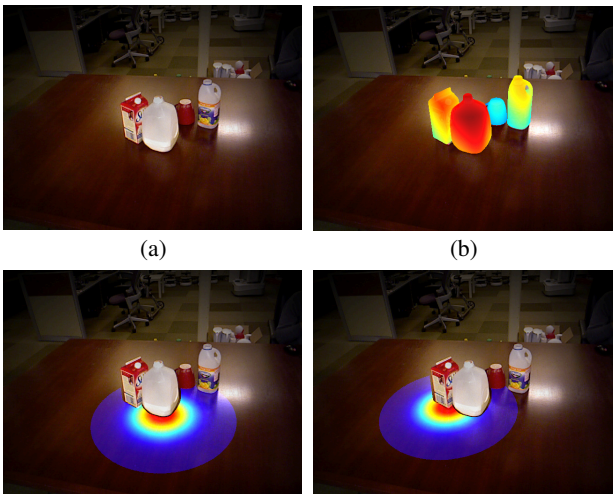


Fig. 1: (a) The input RGB-D scene. (b) The depth image reconstructed by our algorithm superimposed on the input. (c) & (d) High confidence detections from a region-based convolutional neural network for the milk jug and carton. This example shows how D2P can use the hypotheses generated by a discriminative learner in a global search for the best explanation of the scene.

One of the drawbacks of using scene-to-model feature matching or learning methods for multi-object instance detection is that they do not handle the combinatorics of the problem. Typical approaches proceed by obtaining a set of hypotheses for each object in the scene using the aforementioned methods, and then perform some sort of global refinement that takes various geometric constraints into account [3]. A fundamental problem with such methods is that they can fail to produce a feasible global solution, as conflicts between the initial set of hypotheses could be irreconcilable. Moreover, methods that 'train' on single-object instances often have difficulty in scenes where objects are occluded [27], and resort to post-hoc occlusion reasoning to account for missing

features. While training these methods on combinations of objects might help, dataset generation becomes expensive and scales poorly with the number of objects.

In our earlier work, Perception via Search (PERCH) [16], we proposed an analysis-by-synthesis method that uses a tree search decomposition and parallelization to facilitate exhaustive global search for the rendering that best matches the input. However, PERCH is entirely generative and does not leverage the efficiency of fast discriminative algorithms that do perform well when test and training scenes are similar. While we build on PERCH's theme in this work, the primary focus is on demonstrating how learning-based discriminative algorithms can be integrated in a generative search framework, thereby combining the best of both worlds. Our contributions are summarized below:

- Discriminatively-guided Deliberative Perception (D2P), a paradigm for multi-object identification and 3 DoF (x, y, yaw) pose estimation that integrates discriminative learning with generative search. As a particular example, we demonstrate how region-based convolutional neural networks trained on synthetic data can improve performance of exhaustive search on a real-world dataset.
- A lazy multi-heuristic search algorithm for efficient search over rendered scenes, inspired by the lazy weighted A* algorithm [6].
- Notions of completeness and resolution-completeness for multi-object instance detection and pose estimation.

Our evaluations on an existing real-world RGB-D dataset demonstrate how D2P can effectively combine discriminative algorithms with exhaustive search to produce accurate identification and pose estimation. Finally, we provide an open-source implementation of D2P at www.sbpl.net/Software/d2p.

II. Related Work

The availability of economic RGB-D sensors and challenges in 6 DoF pose estimation [7, 12] have brought renewed vigor to research in object instance detection and pose estimation. State-of-the-art approaches for model-based pose estimation broadly fall under two categories: descriptor-based methods (including hand-crafted as well as learned descriptors) and analysis-by-synthesis methods. We juxtapose our proposed D2P paradigm with both camps, and then describe how recent successes of deep learning in the object detection community can help with pose estimation.

Descriptor-based Approaches. Popular object pose estimation approaches are based on either *local* or *global* descriptors [2]. Local descriptor methods (e.g., [11, 24]) first detect and match a set of features between the 3D model and the observed scene, and then perform geometric verification and transformation estimation to obtain the full 6 DoF pose. Global-descriptor pipelines [25, 1, 4, 31, 15] on the other hand use a single-shot procedure to match a pose-preserving decriptor of the object (computed during a training phase) to descriptors computed over object clusters in the scene. Both local and global approaches employ refinement techniques such as Iterative Closest Point (ICP) [5] and Bingham Procrustrean Alignment [9] to fine-tune their pose estimates. We refer the reader to the survey paper by Aldoma et al. [2] for details of local and global descriptor pipelines, and their implementations in the Point Cloud Library [23]. Closely related to global descriptors are template-based methods which precompute [10] or statistically-learn [30] multimodal templates of the object from different viewpoints of the object and perform sliding window search over a test scene.

A fundamental limitation of descriptor and template-based approaches is that they are designed to detect single object instances. When required to detect multiple object instances, such approaches often resort to post-hoc occlusion reasoning and feasibility verification to refine estimates of individual object poses [3]. More importantly, they suffer when key discriminative features become occluded and multiple object instances are present in the scene, as shown by Tejani et al. [29]. On the other hand, the proposed D2P paradigm is at heart a generative approach that accounts for occlusions, thereby overcoming common pitfalls.

Analysis-by-Synthesis. Contrary to descriptor-based methods that rely on matching key features between models and scenes, analysis-by-synthesis or generative methods work on a principle of rendering and verification. This is particularly suited for robotics applications where the 6 DoF camera pose, object models and environment context are often available. A very early work in this area is that of Render, Match and Refine (RMR) by Stevens and Beveridge [28], who use iterative optimization to find a rendering that best matches the input. However, their method is applicable only to 2D images and still dependent on low-level feature extraction for computing similarity between rendered and observed scenes. Krull et al. [14] train a convolutional neural network (CNN) to compare rendered and observed depth images, but their work is restricted to single object instances and does not deal with the combinatorics of multi-object pose estimation. This paper is directly related to and builds upon our earlier work Perception via Search (PERCH) [16], which introduced an efficient tree search formulation of the multi-object localization problem. While PERCH was designed to handle the combinatorics of the multi-object pose estimation problem, it did not leverage any discriminative learning to make global search efficient.

Deep Learning. CNNs have revolutionized the field of object detection in RGB images through their excellent representation learning capabilities [13]. Consequently, recent works in RGB-D object detection have also focused on using deep learning [26, 8] for automatic representation learning. Despite the promise shown, deep learning methods by themselves are ill-suited for multi-object instance detection problems since the required training data is combinatorial in the number of objects. The proposed D2P paradigm, however, strives to use discriminative learners such as CNNs exclusively as heuristics in guiding a global search for the best rendering, thereby shifting responsibility from discrimination to deliberation.

III. PRELIMINARIES

A. Formulation

The problem setup and optimization formulation are similar to those in PERCH [16], which we will re-state here for convenience. We are concerned with estimating the 3 DoF pose (x, y, yaw) of K objects in a single input point cloud, given the 3D models of N unique objects in the scene. We assume $K \geq N$, allowing for multiple copies of a particular instance. This is typical in tabletop manipulation scenarios where objects vary only in translation and yaw with respect to a given 3D model. We further assume that the number of objects K and the 6 DoF camera pose are known a priori, and that the input point cloud contains points only belonging to objects of interest. While the assumptions are indeed strong, they are reasonable for many practical scenarios: for instance, in the Amazon Picking Challenge which replicates warehouse automation, robots are given a task order that lists the set of objects in every shelf bin. Further, in partially controlled settings such as the warehouse, preprocessing is possible to filter sensor data corresponding to extraneous objects such as the shelf. In Sec. VI, we will discuss potential ways to overcome the 3 DoF pose limitation.

Given the setup, multi-object pose estimation is formulated as minimizing a balanced-outlier objective. Following the notation used in [16] (Table. I), let $O_{1:K}$ denote the 3 DoF pose of each of the K objects in the scene, I be the input cloud and R_j the point cloud obtained by rendering $j(\leq K)$ objects, taking occlusions into account. We then seek the minimizer of the following objective:

$$J(O_{1:K}) = \underbrace{\sum_{p \in I} \textsc{Outlier}(p|R_K)}_{J_{observed}(O_{1:K}) \text{ or } J_o} + \underbrace{\sum_{p \in R_K} \textsc{Outlier}(p|I)}_{J_{rendered}(O_{1:K}) \text{ or } J_r} \quad (1)$$

in which $\textsc{Outlier}(p|C)$ for a point cloud C and point p is defined as follows:

$$\textsc{Outlier}(p|C) = \begin{cases} 1 & \text{if } \min_{p' \in C} \|p' - p\|_2 > \delta \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where δ is the sensor noise resolution. Intuitively, this cost function captures how well a rendered scene matches the observed scene in terms of both occupied as well as free space.

B. PERCH Overview

PERCH [16] finds a globally optimal or bounded suboptimal solution of the desired objective 1 over a discretized solution space by casting the joint optimization over object poses as a tree search problem. The tree's root state s_0 is an empty scene with no assignments, and every state down the tree introduces a new object in addition to those already existing in its parent state—i.e, a state s_j in level j of the tree contains a partial assignment of j object poses $\{O_1, O_2, \ldots, O_j\}$. An additional constraint is that the newly added object for a state does not occlude those already present in the parent state. This ensures that the cost of the shortest path from the root state to any state in level K of the tree is equal to the optimum value of 1 over the discretized solution space. The cost of an edge between state s_{j-1} in level j and state s_j in level j is given by

$$c(s_{j-1}, s_j) = \Delta J_r^j + \Delta J_o^j \quad (3)$$

where,

$$\Delta J_r^j = \sum_{p \in \Delta R_j} \textsc{Outlier}(p|I)$$

$$\Delta J_o^j = \sum_{p \in \{I \cap V(O_j)\}} \textsc{Outlier}(p|\Delta R_j) + \textsc{Residual}(j)$$

$$\textsc{Residual}(j) = \begin{cases} \sum_{p \in \{I - V_K\}} \textsc{Outlier}(p|R_K) & \text{if } j = K \\ 0 & \text{otherwise} \end{cases}$$

Given the tree construction and edge cost formulation, PERCH uses Focal-MHA* [17] to find the best path from the root state to any state in the K^{th} level and returns the last state in the found path as the solution to the problem. Focal-MHA* is an informed search algorithm similar to weighted A* (wA*) [19], with a key difference being that it allows for the use of multiple *inadmissible* heuristics in addition to one consistent (and hence admissible) heuristic. It maintains a single priority queue of frontier states sorted in the same order as in wA*, and interleaves expansions of the best state with states chosen greedily by the inadmissible heuristics. Focal-MHA* returns a solution whose cost is no more than a user-chosen factor $w(\geq 1)$ times the optimal solution cost.

An important implementation detail in PERCH is that every time a child/successor state is generated with a new object pose, local-ICP is used to refine the pose of that object to account for discretization artifacts. In more detail, if $\tilde{O}_j$ is the pose of the last added object and $\Delta R_{j,\text{initial}}$ is the point cloud corresponding to $\tilde{O}_j$ (after considering occlusions by existing objects), locally-constrained ICP is used to obtain an adjusted point cloud $\Delta R_{j,\text{adjusted}}$ and corresponding object pose O_j. Then, the successor state is re-rendered with the refined pose O_j to obtain R_j and ΔR_j, which go into the edge cost computation (Eq. 3).

While PERCH uses a tree search formulation to avoid the intractability of joint global optimization, it is still brute-force in many ways since the heuristics it uses (such as the depth-first ordering) are uninformative and do not leverage any discriminative techniques to better guide the search.

TABLE I: Notations

I	The input point cloud
K	The number of objects in the scene
N	The number of unique objects in the scene $(\leq K)$
O	An object state specifying a unique ID and 3 DoF pose
R_j	Point cloud corresponding to the rendering of a scene with j objects $O_{1:j}$
ΔR_j	Point cloud containing points of R_j that belong exclusively to object O_j
$V(O_j)$	The set of points in an admissible (conservative) volume occupied by object O_j, e.g, the volume of the inscribed cylinder
V_j	The union of admissible volumes occupied by objects $O_{1:j}$

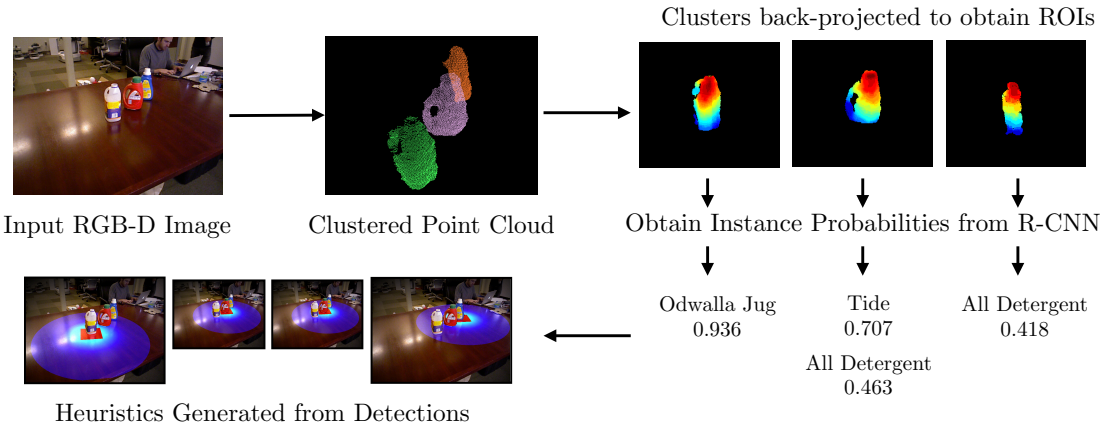


Fig. 2: Discriminative heuristic generation pipeline: First, the point cloud corresponding to the input scene is clustered into K components (for e.g., using PCL's Euclidean cluster extraction) and the points in each cluster are back-projected to obtain ROIs in the depth image. Then, the ROIs are fed to an R-CNN object detector trained on the complete object instance database, after appropriate scaling and colorization. Finally, every high-confidence class prediction for an ROI is converted to a heuristic for global search. In this example, we see that the R-CNN predicts two possible hypotheses for the center ROI, which results in two heuristics being created for that ROI.

IV. Discriminatively-guided Deliberative Perception

A. Discriminative Heuristic Generation

We now present the main contribution of this work—a technique to guide global search using discriminative learners. At a high level, the idea is to obtain a set of hypotheses for every object in the scene and treat each hypothesis as a separate heuristic in the Focal-Multi-Heuristic A* (MHA*) framework [17]. This permits the global search to independently explore different routes down the tree by chaining different hypotheses. For instance, hypothesis 1 might help in selecting state s_1 from level 1 of the tree, while hypothesis 2 could then be used to evaluate all the states in level 2 that were generated as a consequence of expanding s_1. As a result, the search can quickly progress along the optimal route if the hypotheses turn out to be useful, while at all times retaining the ability to backtrack and explore alternative explanations.

While the proposed method is applicable to arbitrary learning algorithms that produce posterior distributions of individual object poses in the scene, we will describe our methodology in the context of object detectors that produce confidence scores for a given bounding box in the depth image, without additional information about 3 DoF pose. Evidently, this is motivated by the availability of successful object detectors from the 2D vision community [13].

Let l denote the label associated with a unique object model, B_i the set of ROIs (bounding boxes) in the depth image and $c(l|B_i)$ the confidence score for object instance l being present in B_i. For every detection with $c(l|B_i) \geq c_{\text{thresh}}$, we generate a heuristic as follows:

$$\bar{p} = \textsc{ProjectToSupportPlane}(\textsc{Centroid}(\{p | p \in B_i\}))$$

$$h(s_j) = \begin{cases} \infty & \text{if } \text{id}(O_j) \neq l \\ 0 & \text{if } \|\bar{p} - T(O_j)\|_p \leq r_{\text{detector}} \\ \|\bar{p} - T(O_j)\|_p & \text{otherwise} \end{cases} \quad (4)$$

where $\|\cdot\|_p$ is the p-norm and $T(O_j)$ is object O_j's center (assuming all models have been preprocessed such that the z-coordinate of their origins have been set to the height of the supporting surface), ignoring the orientation. Essentially, every heuristic acts as "do not care" when the last added object is different from the detection's label and equally prefers all states within the r_{detector} p-ball of the detection's centroid if the labels match. Figure 2 illustrates the heuristic generation process. Note that we could have multiple hypotheses for the same ROI (e.g., when the bounding box covers multiple objects in the scene), and the onus falls on the search to resolve conflicts and produce a globally feasible solution.

B. Lazy Focal-MHA*

One of the most computationally expensive components of PERCH is the rendering of successor scenes when expanding a state in the tree. This is aggravated by the need to render each scene twice (first to obtain the point cloud that is used for ICP adjustment, and the second post-ICP to get the point cloud on which the edge-cost is computed). We propose two optimizations to accelerate this process: the first is caching of first-level states (states with single objects) to quickly produce depth images for multi-object states in deeper levels of the tree, and the second, a lazy evaluation (i.e, postponing exact evaluation until necessary) of edge costs. Together, they completely eliminate the need to render the successor states for an expanded parent. Instead, the only time a state is fully rendered is when it is about to be expanded during the search. We discuss these optimizations in more detail:

Depth Image Caching. Upon expanding the root node of the tree, successor states corresponding to all poses of individual objects are rendered[1]. We cache depth images $D(O_j)$ corresponding to individual object states O_j, and then reconstruct the depth image for a multi-object state comprising of objects $O_{1:k}$ simply by taking the element-wise minimum

[1]In practice, several poses can be pruned before rendering by requiring an assigned object to 'explain' at least one point in the input point cloud.

of depth images $D(O_1), D(O_2), \ldots, D(O_j)$. This eliminates the need to render multi-object successors upon expanding a state in the tree.

Lazy MHA*. Depth-image caching by itself however, is not sufficient to accelerate successor generation. This is because the point cloud corresponding to the newly rendered object goes through ICP refinement and subsequent re-rendering before the edge-cost can be evaluated. A similar bottleneck exists in heuristic search-based motion planning, where expanding a graph state requires time-consuming collision checking of the edges. This was addressed by Cohen et al. [6] in their lazy weighted A* algorithm. The key idea is that if we have a mechanism to inexpensively compute *admissible* estimates of the edge cost, then weighted A* search can simply use these proxy costs while inserting states into the frontier and look up the true cost only when a lazily evaluated state is about to be expanded. By using these admissible estimates (i.e, the estimated edge cost is lesser than or equal to the true cost), lazy weighted A* retains theoretical properties of bounded suboptimality. We apply the same idea to Focal-MHA*, albeit with minor differences. Since Focal-MHA* interleaves admissible expansions with expansions from inadmissible heuristics, we require that the true edge cost be evaluated any time it is about to be expanded, irrespective of whether it was chosen by the admissible heuristic, or an inadmissible one. This ensures bounded suboptimality of the solution returned by Focal-MHA* (the proof follows a similar structure to that of lazy weighted A*, but is omitted here for simplicity).

Next, we describe how a lazy admissible estimate of the edge cost can be obtained without rendering the successor state. Let s_j be a newly generated successor state of s_{j-1} with O_j as the last introduced object, and ΔR_j be the point cloud corresponding to the visible portion of object O_j given the other objects in s_j. The lazy cost of the edge to s_j is computed as follows:

1) Obtain the depth image corresponding to s_j by composing the cached depth image of its parent (which exists by induction) with the cached depth image of O_j.
2) The differential partial cloud ΔR_j of the newly introduced object is subject to ICP refinement, resulting in R'_j.
3) Points in $\Delta R'_j$ that are self-occluded and occluded by other objects in s_j are removed to obtain $\Delta \tilde{R}_j$. Removal of self-occluding points is done by projecting all points in R'_j to the depth image and retaining only the minimum depth for each pixel when multiple points project to the same one. Points occluded by existing objects in s_j are similarly removed by taking the element-wise minimum of the re-projected depth image with the cached depth image of the parent state. Finally, $\Delta \tilde{R}_j$ is obtained by unprojecting the depth image pixels corresponding to O_j, following the above process.
4) The lazy edge cost is then computed as

$$\tilde{c}(s_{j-1}, s_j) = \sum_{p \in \Delta \tilde{R}_j} \textsc{Outlier}(p|I) \tag{5}$$

Theorem 1. *Lazy estimates of the edge cost obtained by the above procedure are guaranteed to be admissible estimates of the true edge cost.*

Proof (Sketch): Let s_j be the considered successor state of s_{j-1} and $\Delta R_{j,\text{true}}$ the differential point cloud corresponding to object O_j. By construction, we have $\Delta \tilde{R}_j \subseteq \Delta R_{j,\text{true}}$. Intuitively, re-rendering an object after ICP refinement will only introduce new unseen portions of the object, while the existing parts continue to be visible or become self-occluded/occluded by other objects. The true cost $c(s_{j-1}, s_j)$ is given by

$$\begin{aligned}
c(s_{j-1}, s_j) &= \Delta J_r^j + \Delta J_o^j && \text{Eq. 3} \\
&\geq \Delta J_r^j && \because \Delta J_o^j \geq 0 \\
&= \sum_{p \in \Delta R_{j,\text{true}}} \textsc{Outlier}(p|I) \\
&\geq \sum_{p \in \Delta \tilde{R}_j} \textsc{Outlier}(p|I) && \because \Delta \tilde{R}_j \subseteq \Delta R_{j,\text{true}} \\
&= \tilde{c}(s_{j-1}, s_j) && \square
\end{aligned}$$

C. Completeness

Finally, we introduce a notion of completeness for the multi-object instance detection and pose estimation problem:

Definition 1 (Completeness)**.** *An algorithm for multi-object pose estimation of K objects is* complete *if it returns* any *feasible solution (i.e, a solution that contains guaranteedly collision-free pose estimates for all K objects) when one exists, and correctly identifies in finite time that no solution exists otherwise.*

This definition mirrors the notion of completeness in motion planning. We also say an algorithm is resolution-complete if it satisfies the above requirements in a smaller solution space obtained after discretization. D2P and PERCH are both resolution-complete for the chosen discretization in the absence of ICP refinement. This follows from the completeness property of Focal-MHA*. When we do use ICP refinement, the algorithms are still resolution-complete, but in a solution space that itself depends on the ICP refinement instead of the chosen discretization. We will discuss the performance tradeoff that arises out of larger ICP-basins versus finer discretization in the evaluation section (Sec. V-C).

We stress upon the notion of completeness since popular approaches to this problem proceed by obtaining individual hypotheses for each object and then performing a global refinement [3], which leads to a restricted solution space and hence algorithm incompleteness.

V. Evaluation

We evaluate D2P on the real-world occlusion dataset introduced by Aldoma et al. [2]. We choose this dataset for two major reasons: first, it satisfies the assumptions we make in this work—the input RGB-D scene can be preprocessed such that the resulting point cloud contains points only belonging to objects of interest. Second, it features an interesting mix of scenes where objects are occluded as well as occur in

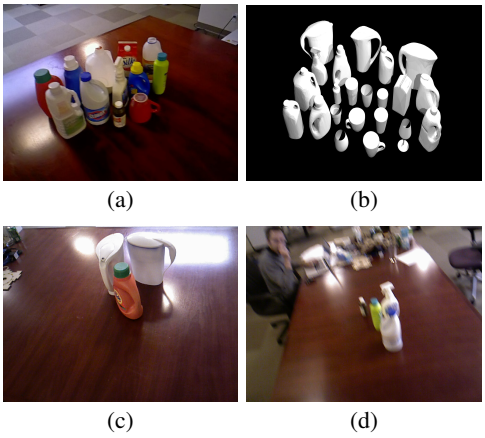


Fig. 3: (a) and (b) A subset of objects in the RGB-D occlusion dataset. (c) and (d) Representative test scenes from the dataset.

isolation, thereby allowing us to demonstrate the complementary strengths of deliberation and discrimination. The dataset contains 3D models of 36 household objects and 22 RGB-D scenes with 80 object instances in total, ignoring one non-compatible scene which fails to satisfy our 3 DoF pose assumption.

A. D2P Implementation

Parameters. Table II lists all the parameters used in D2P. Unless otherwise specified in particular experiments, we use the following values: $\delta = 0.003$ m, $dx = dy = 0.1$ m, $d\theta = 22.5°$, $w = 10$, max_icp_iter $= 20$, num_procs $= 40$. We perform local ICP adjustments for newly added objects by constraining ICP to match correspondences only if they are within a distance of $dx/2$. All experiments are performed on a single Amazon AWS m4.10x instance with 40 virtual cores, using MPI parallelization to compute edge costs for successor states in parallel.

TABLE II: D2P Parameters

δ	Sensor noise resolution used in Eq. 2
$dx, dy, d\theta$	The discretizations for (x, y, θ) coordinates
w	Suboptimality factor used by Focal-MHA*
max_icp_iter	Max. number of ICP iterations for refinement
$r_{\text{detector}}, c_{\text{thresh}}$	Heuristic-generation parameters (Eq. 4)
num_procs	Number of processors used for parallelization

Deep-Learning Heuristics. We generate heuristics for Focal-MHA* using an object detector as described in Sec. IV-A. We leverage a state-of-the-art implementation of region-based convolutional neural networks: Faster-RCNN [20]. A common trend in the 2D object detection community is to use networks pre-trained on large training datasets such as Imagenet [22] to initialize training on a custom dataset. We have two major concerns to address: the generation of training data for the 36 object models in our dataset, and a method to encode depth-images as 3-channel images—the input format used by available deep neural network implementations. We generate our training data by synthetically rendering every object in isolation from camera poses sampled uniformly on concentric cylinders around the object. We also create duplicates of the generated scenes by a) adding artificial noise—treating a randomly chosen 15% of pixels in the rendered image as no-returns, and b) introducing occlusions in the form of a circle placed at a random valid image pixel. The radius of these circles is chosen to be one-third of the rendered object's bounding box. Finally, we obtain 108864 training images in total, with each annotated by a bounding box and label for the object present in it. For encoding depth-images as 3-channel images, we follow the method adopted by [8] who apply a jet-coloring of the $[0-255]$ rescaled depth-image. While there is no theoretical justification for this process, the intuition is that jet-color maps encode discontinuities in depth as discontinuities in color, making them suitable for networks pre-trained on RGB images. We use the ZF network architecture [20] by modifying the final fully-connected (FC) layer train to span 36 object clases, and use the default 4-stage Faster-RCNN training settings. The training takes $\sim$ 30 hours on an Amazon g2.2xlarge GPU-enabled instance. For generating heuristics for the search from object detections in the depth image, we use $c_{\text{thresh}} = 0.2$ (confidence scores are normalized to [0,1]), $r_{\text{detector}} = 0.1$ and $p = 1$ for the norm in Eq. 4. Note that we are able to use a high recall threshold because spurious detections simply translate to uninformative heuristics for the search, without affecting the final solution quality. However, misleading heuristics could have a negative impact on the time taken to find a solution. In addition to the heuristics generated by the deep-learning procedure, we use one additional depth-first heuristic described in PERCH: $h_{depth}(s) = K - |s|$, where $|s|$ is the number of objects in state s. This serves to prefer expanding states deeper in the tree that are closer to a potential goal state. Finally, the consistent heuristic used by Focal-MHA* is the trivial zero heuristic.

B. Baseline Implementations

We compare the performance of D2P with PERCH [16], OUR-CVFH [4] and a brute-force ICP baseline described in [16]. We configure PERCH to use the same parameters as D2P where applicable, and include lazy edge-cost evaluation. Strictly speaking, lazy edge evaluation is a contribution of this work, however our goal is to study how discriminative guidance can help global search.

OUR-CVFH. Oriented, Unique and Repeatable Clustered Viewpoint Histogram (OUR-CVFH) is a state-of-the-art global descriptor specifically developed for providing robustness to partial occlusions. During training phase, OUR-CVFH decomposes objects into stable surfaces and computes separate view-dependent descriptors for each surface rather than one view-dependent descriptor for the entire object, so that descriptor-matching can be robust to partial occlusions and missing surfaces. We generate the training point-cloud database by rendering 642 different views of each object model, sampled uniformly on the viewsphere. We then use the default Point Cloud Library implementation of global descriptor pipelines to

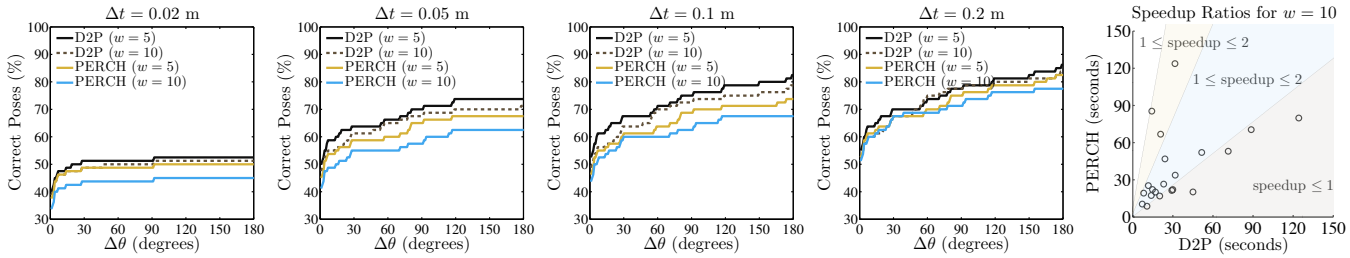


Fig. 4: The first four plots show the percentage of correct poses produced by D2P and PERCH for suboptimality bounds of 5 and 10, where correctness is defined as having translation error within Δt and rotation error within $\Delta\theta$. The discriminative heuristics used by D2P help produce a larger number of correct poses within the given time limit of 5 minutes, for identical suboptimality factors. The scatter plot on the right shows the time taken by D2P and PERCH for every scene, with different shaded regions representing distinct speedup intervals.

upsample all point clouds to the 3 mm Kinect sensor resolution and compute OUR-CVFH descriptors for each training point cloud. Since OUR-CVFH operates by clustering the input cloud into object groups and then detecting the pose of each one, it could result in producing too many or too few object estimates due to under or over-clustering (e.g., when objects are touching each other). Here, we use information about the number of objects in the scene (K) to create the following procedure: first, we take the $\max(K, \#\text{detected clusters})$ largest clusters in the scene and compute the descriptor distance for each cluster to every object we know exists in the scene. Second, we solve a min-cost matching problem over the descriptor distance scores to obtain an assignment that has exactly one estimate for every object. Finally, since we know that the models vary only in 3 DoF, the pitch and roll parts of the 6 DoF pose estimates produced by OUR-CVFH are set to 0, with a final ICP refinement for each object.

Brute Force ICP. A strawman ICP approach to the multi-object instance detection problem is to perform a sliding-box ICP for each object sequentially and take the best fit for each during the process. Since the order in which the objects are chosen matters (e.g., if object A occupied a location first, then object B has fewer locations to be assigned to), the entire process could be repeated for all possible orderings of the objects. Note that this does not involve rendering at all, and uses the point cloud of the full object model at every step for ICP refinement. We call this baseline BF-ICP.

C. Results

Comparison with Baselines. We measure accuracy of an algorithm by counting the number of objects that fall within a given error bound. Specifically, an estimated object pose (x, y, θ) is declared 'correct' if $\|(x, y) - (x_{\text{true}}, y_{\text{true}})\|_2 < \Delta t$ and $\text{SHORTESTANGULARDIFFERENCE}(\theta, \theta_{\text{true}}) < \Delta\theta$. The second portion is ignored for rotationally symmetric objects. Figure 4 compares the performance of D2P with PERCH configured with identical parameters (including lazy edge evaluation), but for the discriminative heuristics. We set an upper limit of 5 minutes for each scene and take the best solution discovered thus far if time runs out. While all experiments are done on m4.10x AWS instances, the objector detector outputs for D2P alone are precomputed for all scenes on an Amazon AWS g2.2xlarge GPU instance (which takes ~ 0.2 seconds per scene). The first four plots show the cumulative number of correct poses as $\Delta\theta$ is increased, for a fixed value of Δt. Two trends are evident: a) D2P dominates PERCH consistently, and b) lower suboptimality factors (w) produce more correct poses than higher ones. The latter is expected from the behavior of Focal-MHA*; however it comes at the price of a longer time to find a solution. The final plot compares the run times of D2P and PERCH for every scene in the dataset, for a common suboptimality bound of $w = 10$. As we expect to see, D2P has a speedup over PERCH for majority of the scenes. An interesting observation is that there are also few cases where D2P is slower than PERCH. We find this to occur either in scenarios where the heuristics are misleading (e.g., false positives from the RCNN detector), or when there are too many heuristics (due to very high recall) resulting in significant overhead for Focal-MHA*. We believe both these problems can be alleviated to some extent by following a technique similar to the one of Phillips et al. [18], where the "progress" made by each heuristic is monitored to intelligently schedule computation to each heuristic, rather than following the naive round-robin scheme used by Focal-MHA*.

Figure 5 depicts an identical comparison to OUR-CVFH and BF-ICP. We follow the same methodology as for the comparison with PERCH, and give D2P a maximum time limit of 5 minutes to find a solution. OUR-CVFH being a descriptor-matching method requires no time limit, whereas BF-ICP is provided sufficient time to exhaust all possible orderings of the objects. The results show that D2P consistently dominates the baselines, while showing most gain for strict error measurement criteria. Although BF-ICP performs an exhaustive search over all possible orderings of the object, the lack of any intermediate rendering to account for self-occlusions and occlusions by other objects inhibits its performance. The mean computation time per scene for BF-ICP, OUR-CVFH and D2P ($w = 5$) were 104.35, 5.02, and 139.74 seconds respectively. BF-ICP required no training, while OUR-CVFH needed $\sim$ 14 hours to render the objects from different viewpoints and build the descriptor database (Sec. V-B). The training time for D2P depends on the discriminative learner used, which in our particular implementation is the R-CNN. As noted in Sec. V-A, the training time for the ZF R-CNN was $\sim$ 30 hours.

Utility of Lazy Edge Evaluations. We next study how useful lazy edge cost evaluations are, with regard to the branching factor of the tree and the amount of parallelization

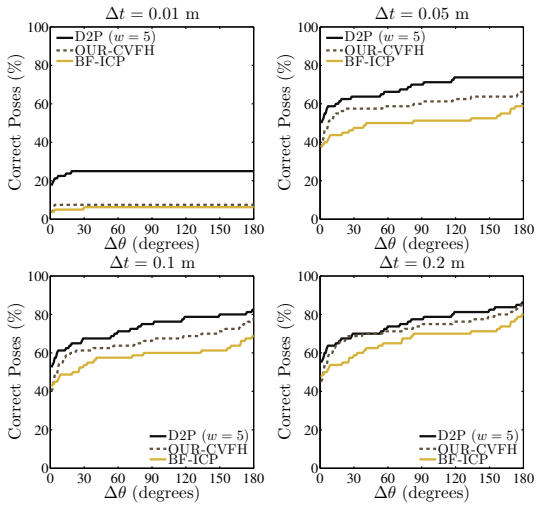
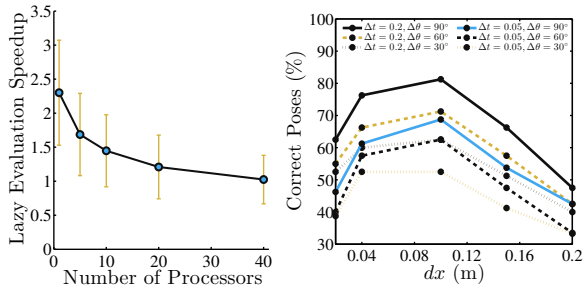


Fig. 5: Comparison of D2P with OUR-CVFH and BF-ICP for different correctness criteria. D2P outperforms the baselines consistently, with the margin being larger for stricter correctness conditions.



(a) (b)

Fig. 6: (a) Speedup obtained by lazy evaluation of edge costs as a function of parallelization. The values are mean speedups across every input scene, with the error bars representing one standard deviation. (b) Performance as a function of the translation-discretization, with each trace corresponding to a specific correctness criterion.

available. Figure 6a plots the mean speedup of lazy D2P over non-lazy D2P for a varying number of processors available, setting $w = 10$. We observe that lazy evaluation is most useful when parallelization is limited and vice versa. If t were the time required to compute the true cost of an edge, $t_{\text{lazy}}(<< t)$ the time to compute the lazy edge cost, E the number of expansions required to find a solution for both lazy and non-lazy variants, N the number of processors used to compute edge costs in parallel, and b the branching factor for every tree state, then non-lazy D2P would take $t(b/N)E$ time to return a solution, whereas lazy D2P would take $tE + t_{\text{lazy}}(b/N)E$ to return a solution. Clearly, the benefits of lazy D2P are pronounced when the effective branching factor (b/N) is large. While we vary effective branching as a function of N in Fig. 6a, a similar trend would show if we vary b instead, e.g., using a finer discretization or more objects in the scene.

Discretization vs. ICP Tradeoff. A key implementation detail is that of local ICP refinement for every newly added object to a successor state. In our implementation, we restrict ICP refinement to only use correspondences that are within $dx/2$ when iteratively estimating the 3 DoF transformation, to keep ICP 'local' to the grid cell at which an object is placed by the search. This immediately introduces the following tradeoff: for coarse discretizations, ICP would have a larger basin of attraction to get to the best fit; the price being that the initial object locations generated by the search might altogether miss small objects. While using a finer resolution might help, it comes at the cost of a larger branching factor and restricted locality for ICP. This tradeoff is captured in Fig. 6b, which suggests a sweet-spot somewhere in the middle. More generally, our future work here entails using adaptive-resolution search as well as smarter local refinement techniques to combat the aforementioned problems. This experiment uses a maximum time limit of 10 minutes, to accommodate large branching factors resulting from finer discretizations.

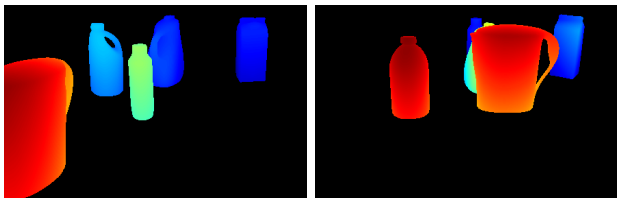


Fig. 7: Synthetic example demonstrating the complementary strengths of discriminative and deliberative methods.

Synthetic Example We conclude with a synthetic example that reiterates the complementary strengths of discriminative guidance and systematic search. Figure 7 shows two scenes containing the same set of 5 objects in different configurations. In the first scene, the objects are mostly isolated and non-occluded. Search by itself takes a long time to obtain a solution (over 13 minutes for $dx = 0.1$ and 8-core parallelization) since it has no informative heuristic. However, by using the guidance from the discriminative R-CNN which correctly identifies all objects, the time to obtain a solution is reduced to just under 7 minutes. The second scene is more complex and features severe occlusions. Here, D2P manages to reconstruct the complete scene although the R-CNN correctly identifies only the 3 non-occluded objects.

VI. Conclusion

We presented a novel paradigm, D2P, for integrating discriminative algorithms with global search for the task of multi-object identification and 3 DoF localization. D2P achieves this by treating the predictions of discriminative learners as heuristics in a multi-heuristic search framework. While our results indicate significant improvement in performance over state-of-the-art methods, several improvements remain to be made. For example, one future direction of work is to handle full 6 DoF pose estimation by using constraints from physics to eliminate large portions of the search space. Other directions include the use of RGB data and optimizations in the search backend (e.g., through partial and parallel node expansions) for faster solutions. Finally, our implementation of D2P is available at www.sbpl.net/Software/d2p.

Acknowledgment

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

References

[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 *ICCV Workshops*. 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, Luigi Di Stefano, and Markus Vincze. A Global Hypotheses Verification Method for 3D Object Recognition. In *ECCV*, pages 511–524. 2012.

[4] 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. In *DAGM*, 2012.

[5] Yung Chen and Gérard Medioni. Object Modeling by Registration of Multiple Range Images. In *ICRA*, 1991.

[6] Benjamin Cohen, Mike Phillips, and Maxim Likhachev. Planning Single-arm Manipulations with N-Arm Robots. In *Proceedings of Robotics: Science and Systems*, 2014.

[7] Nikolaus Correll, Kostas E. Bekris, Dmitry Berenson, Oliver Brock, Albert Causo, Kris Hauser, Kei Okada, Alberto Rodriguez, Joseph M. Romano, and Peter R. Wurman. Lessons from the Amazon Picking Challenge. *arXiv preprint arXiv:1601.05484*, 2016.

[8] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. Multimodal Deep Learning for Robust RGB-D Object Recognition. In *IROS*, 2015.

[9] Jack Glover and Sanja Popovic. Bingham Procrustean Alignment for Object Detection in Clutter. In *IROS*, pages 2158–2165. IEEE, 2013.

[10] 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 *ACCV*, pages 548–562. 2013.

[11] Andrew E Johnson and Martial Hebert. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *PAMI*, 21(5):433–449, 1999.

[12] Tae-Kyun Kim, Vincent Lepetit, Carsten Rother, Jiri Matas, Ales Leonardis, and Rigas Kouskouridas. 1st International Workshop on Recovering 6D Object Pose. In *ICCV*, 2015.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet Classification with Deep Convolutional Neural Networks. In *NIPS*, pages 1097–1105, 2012.

[14] Alexander Krull, Eric Brachmann, Frank Michel, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Learning Analysis-by-Synthesis for 6D Pose Estimation in RGB-D Images. In *ICCV*, 2015.

[15] 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.

[16] Venkatraman Narayanan and Maxim Likhachev. PERCH: Perception via Search for Multi-Object Recognition and Localization. In *ICRA*. IEEE, 2016.

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

[18] Mike Phillips, Venkatraman Narayanan, Sandip Aine, and Maxim Likhachev. Efficient Search with an Ensemble of Heuristics. In *IJCAI*, pages 784–791, 2015.

[19] I. Pohl. First Results on the Effect of Error in Heuristic Search. *Machine Intelligence*, 5:219–236, 1970.

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. In *NIPS*, pages 91–99, 2015.

[21] Lawrence Gilman Roberts. *Machine Perception of Three-Dimensional Solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[23] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point Cloud Library (PCL). In *ICRA*, pages 1–4. IEEE, 2011.

[24] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. In *ICRA*, pages 3212–3217. IEEE, 2009.

[25] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram. In *IROS*. IEEE, 2010.

[26] Max Schwarz, Hannes Schulz, and Sven Behnke. RGB-D Object Recognition and Pose Estimation Based on Pre-trained Convolutional Neural Network Features. In *ICRA*. IEEE, 2015.

[27] Mark R Stevens and J Ross Beveridge. Integrating Graphics and Vision for Object Recognition. volume 589. Springer Science & Business Media, 2000.

[28] Mark R Stevens and J Ross Beveridge. Localized Scene Interpretation from 3D Models, Range, and Optical Data. *Computer Vision and Image Understanding*, 80(2):111–129, 2000.

[29] Alykhan Tejani, Danhang Tang, Rigas Kouskouridas, and Tae-Kyun Kim. Latent-class Hough Forests for 3D Object Detection and Pose Estimation. In *ECCV*, pages 462–477. 2014.

[30] Paul Wohlhart and Vincent Lepetit. Learning Descriptors for Object Recognition and 3D Pose Estimation. In *CVPR*, pages 3109–3118, 2015.

[31] Walter Wohlkinger and Markus Vincze. Ensemble of Shape Functions for 3D Object Classification. In *ROBIO*, pages 2987–2992. IEEE, 2011.