Globally Optimal Object Pose Estimation in Point Clouds with Mixed-Integer Programming

Gregory Izatt[1], Hongkai Dai[2] and Russ Tedrake[1,2]

Abstract Motivated by the limitations of local object trackers, we present a formulation of the underlying point-cloud object pose estimation problem as a mixed-integer convex program, which we efficiently solve to optimality with an off-the-shelf branch and bound solver. We show that reasoning about object pose estimation in this way allows natural extension to point-to-mesh correspondence, multiple simultaneous object pose estimation, and outlier rejection without losing the ability to obtain a globally optimal solution. We probe the extent to which rich problem-specific formulations typically tackled with unreliable nonlinear optimization can be rigorously treated in a global optimization framework to overcome the limitations of other global pose estimation methods.

1 Introduction

The robotic perception community has placed significant emphasis on designing and improving perception modules for estimating the poses of objects in a scene. These modules have enormous value for autonomous systems, in that they reduce extremely high-dimensional sensor inputs to compact and semantically loaded object state that can be consumed by a broad range of motion planners and robot controllers.

Here, we are concerned with systems for estimating object poses from point cloud information, e.g. from increasingly ubiquitous RGB-D cameras. Myriad techniques perform pose estimation without an initial guess, e.g. via sampling [11, 24, 28], feature extraction [7, 17, 34, 35, 38], template matching [14], shape descriptors [1], and direct machine learning [36]. However, because of the scale of the sensory data and the difficulty of the global optimization, few of these techniques run in real time, and those that do can not make claims concerning reliability and convergence to global optimality. Given a reasonable initial guess, another broad class of pose tracking techniques perform real-time object tracking. These techniques have grown

[1] Computer Science and Artificial Intelligence Lab, MIT [2] Toyota Research Institute
[*gizatt@mit.edu, hongkai.dai@tri.global, russt@mit.edu*]

extremely mature, boasting support for multiple articulated objects [31], deformable objects [32], contact [20], and tactile sensing [16, 18, 30].

In this work, we present a study of the core optimization problem that underlies many of these techniques. We show that the pose estimation problem for point clouds can be viewed through the lens of mixed-integer programming, and that doing so leads to a problem formulation permitting optimization to certifiable approximate global optimality via branch and bound search. This formulation is written in a general form that is extensible to handle explicit outlier rejection and multiple models, and can consume the output of other local and global pose estimation algorithms as seeds to accelerate the global search – and in doing so, verify the optimality of the output of those other algorithms.

2 Problem Formulation

We focus on an instance of the *point-cloud object pose estimation problem* that involves finding the best configuration of a rigid-body model to explain the data available from a sensed point cloud. The model configuration is parameterized by the rotations $R \in SO(3)$ and translations $T \in \mathcal{R}^3$ of each rigid body. The point cloud sensor samples a set of N_s points $S = \{s_i\}$ from the geometry of the world. Many techniques represent the model as a collection of N_m point features, which leads to an optimization penalizing a norm (shown here in the single-object case):

$$\begin{aligned} &\min_{R,T,C} \sum_{i \in [1,N_s]} \left\| Rs_i + T - m_{C(i)} \right\|, \\ &C(i) = \operatorname*{argmin}_{j \in [1,N_m]} \left\| Rs_i + T - m_j \right\|, \end{aligned} \tag{1}$$

where $C(i)$ corresponds each scene point to the closest model feature according to the desired norm.

Objectives like this one are reflected in many of the pose estimation techniques in the literature. Key differences between these techniques lie in the model representation and distance function used, the method of optimization, and further additions to the objective beyond optimization of just a distance function.

A critical feature of this problem is that the correspondences C and transformation $\{R, T\}$ are each independently sufficient to specify a solution to this problem. Given the correspondences, the optimal transformation can be computed in closed form [9]. Given the transformation, correspondences can be backed out if desired via, e.g., closest point lookups on the model. The famous Iterative Closest Point (ICP) algorithm, from which many ob-

ject trackers are inspired, performs Expectation Maximization by alternating between solving these two problems [3, 5].

3 Related Work

As described in Section 1, numerous techniques have been developed to estimate object poses without an initial guess. However, in order to make the global optimization more tractable, most of these techniques rely on methods like stochastic sampling and downsampling via feature extraction, and additionally may make assumptions concerning the quantity of outliers and prior knowledge of object segmentations. At a high level, several broad classes of approaches have been used to attempt global optimization of the point cloud pose estimation objective directly on the complete raw point cloud, including semidefinite programming (SDP) relaxation, and branch and bound search.

SDP relaxation of the rotation and correspondence constraints constrain R to be within a convex hull of $SO(3)$, and allow a continuous relaxation of C [2]. This relaxation transforms the difficult nonlinear problem to a much easier convex one. This technique has proven very powerful for solving the Procrustes Matching (PM) problem, [22], and similar SDP relaxation can be applied to the pose estimation problem with fixed correspondences (i.e. an alignment problem) [4]. Their method boasts tightness up to a quantified noise threshold, and is demonstrated aligning 800 points across 30 overlapping point clouds. SDP relaxations have also proven powerful for performing large-scale, noisy pose graph optimization over $SE(3)$ [29]. However, we have found that these relaxations are difficult to apply directly to the point-cloud pose estimation problem, because the SDP relaxation of $R \in SO(3)$ is very loose and introduces trivial solutions to Eq. 1.

Other methods perform branch and bound over the space of rotations [13, 19, 27], or rotations and translations [37]. The latter – GO-ICP – accomplishes our broad goal of providing globally-optimal pose estimates, but it does not explicitly reason about correspondences. This manifests itself most clearly in the handling of outliers: a user of GO-ICP must specify the expected fraction of outliers ahead of time, and setting this parameter incorrectly can lead to invalid results. Other techniques take direct advantage of the property that it is easy to detect inconsistencies in small sets of correspondences in order to prune branches in the search tree [10, 12]. Recent work has also demonstrated a custom search strategy over object arrangements that is aware of the order of object occlusion and its effect on the independence of objects an alignment objective [25]. This work explicitly reasons about outliers, but has not been scaled beyond planar object arrangements.

The transform and correspondence information are tightly coupled in the pose estimation problem. Thus, a formulation that reasons about point correspondences and model transformations *simulataneously* stands to benefit

from this interplay. We present such a formulation based on mixed-integer programming [26]. While even the restricted class of mixed-integer linear programs (MILPs) is itself NP-hard, mixed-integer programs (MIPS) that are convex in their continuous and integer variables are amenable to branch and bound search that can be very efficient, given the right problem structure. These algorithms are implemented by powerful off-the-shelf solvers capable of solving problems with millions of variables and constraints [15].

4 Mixed-Integer Problem Formulation

Our formulation of this problem uses a generalized mesh model to represent the objects, for the reason that the mesh model is significantly more compact than a traditional sampled point model. Given a model defined by N_m vertices and N_f faces, where each face is defined as an affine combination of a subset of coplanar vertices, as well as

- scene points $S = \{s_i\}, i \in [1, N_s]$,
- model vertices $M = \{m_j\}, j \in [1, N_m]$,
- a binary face membership map $F \in \{0,1\}^{N_m \times N_f}$,

the generic pose estimation problem is equivalent to finding a rotation matrix R, a translation matrix T, a combination matrix $C \in \mathcal{R}^{N_s \times N_m}$, and a face correspondence matrix $f \in \{0,1\}^{N_s \times N_f}$ that satisfy the following.

$$
\begin{aligned}
\underset{R,T,C,f}{\text{minimize}} \quad & \frac{1}{N_s} \sum_{i \in [1,N_s]} \left\| Rs_i + T - MC_i^T \right\|_2^2 \\
\text{subject to} \quad & R \in SO(3), \\
& \sum_{j \in [1,N_m]} C_{i,j} = 1, \ \forall i, \\
& \sum_{k \in [1,N_f]} f_{i,k} = 1, \ \forall i, \\
& 0 \leq C_{i,j} \leq F_j f_i^T, \ \forall i, j, \\
& f_{i,j} \in [0,1], \ \forall i, j.
\end{aligned}
$$

C_i and f_i are the i^{th} rows of C and f, and F_j is the j^{th} row of F. The affine combination coefficient for the i^{th} scene point and j^{th} model point $C_{i,j}$ is constrained to be inactive unless one of the faces for model point j is active according to its face map F_j and the face selection f_i. Scene points can only correspond to a single model face. Note that, save the constraint $R \in SO(3)$, this problem is already a mixed-integer convex (quadratic) program. In the next section, we describe how to approximate the constraint $R \in SO(3)$ to completely express this problem in a mixed-integer convex way.

4.1 Mixed-Integer Linear Approximation of $R \in SO(3)$

We approximate the $R \in SO(3)$ constraint with a piecewise-convex outer approximation in the spirit of the McCormick Envelope [23]. For each member of the rotation matrix $R_{i,j}$, we introduce new binary variables to assign $R_{i,j}$ to one of N_k partitions of $[-1, 1]$. These binary variables are used to activate region-specific constraints approximating the constraints implied by $R \in SO(3)$, based on the formulation in [6]. The constraints approximated are $R^T R = I$ and $R_1 \times R_2 = R_3 \iff \implies det(R) = +1$ for the sets of rows and columns $\{R_1, R_2, R_3\}$ of R.

For example, if we denote a column of R as $[x, y, z]^T \in \mathcal{R}^3$, $R^T R = I$ implies that this vector should have a unit length and hence lie on the surface of the unit sphere. The partitioning of x, y, and z can be geometrically interpreted as cutting the surface of the unit sphere by planes parallel to either xy, xz or yz planes, as shown in Fig. 1. The intersection between the planes and the sphere partitions the surface of the sphere into small regions (Fig. 2). For each surface region, we can readily compute a convex polytope $A[x\ y\ z\ ]^T \leq b$ containing the region (Fig. 3). This linear constraint is activated by the binary variables denoting in which interval the variables x, y, z lie. Similar constructions are used to approximate the remaining orthogonality and cross-product constraints.

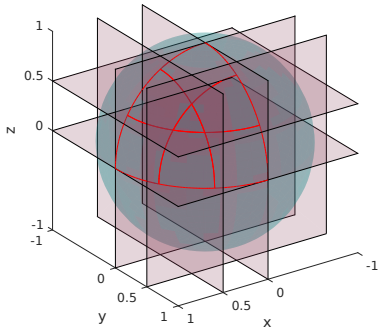


Fig. 1: We cut the first octant of the sphere with regularly spaced planes normal to each axis. The intersections between the planes and the sphere (red arcs) partition the surface of the sphere into smaller regions.

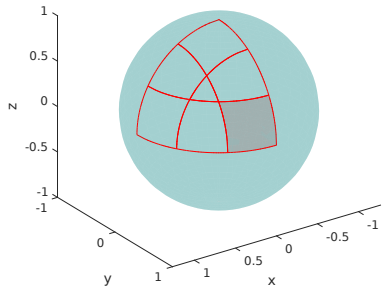
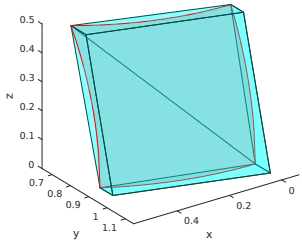


Fig. 2: The first octant of the sphere surface after partition in Fig. 1. The shaded surface region is one of the partitions.



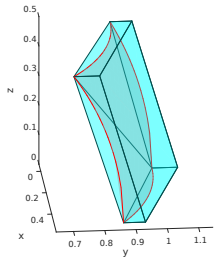


Fig. 3: The polytope $A[x\ y\ z]^T \leq b$ (light blue region) containing the shaded surface region in Fig. 2, viewed from two perspectives. We relax the constraint that the vector is on the shaded surface area, to that the vector lies within the polytope.

5 Extensions

5.1 Handling Outliers

Correct outlier handling is critical for object pose estimation algorithms, as point clouds in the wild invariably include unmodeled points from nearby objects and support surfaces in the scene. We extend this formulation to allow scene points to be explicitly classified as outliers.

We first switch from the L-2 to the L-1 norm in our error metric, so that we can include the distance to each point in the set of linear constraints. We introduce an intermediate variable ϕ_i for each scene point s_i storing the L-1 distance from s_i to the matched point on the model. Additional slack variables $\alpha^i \in R^3$ are introduced to implement 3 absolute values within the L-1 norm for each scene index i. We bound ϕ_i with a constant *maximum allowed L-1 distance* ϕ_{max} as a threshold (and penalty) for classifying points as outliers. Finally, we add a new binary variable o_i for each scene point indicating that that scene point is being considered an outlier.

Extending the mesh-model MIP formulation from Section 4, we now solve the MILP (for large $\mathbb{M}$):

$$
\begin{aligned}
\underset{R,T,C,f,\phi,\alpha,o}{\text{minimize}} \quad & \min \frac{1}{N_s} \sum_{i \in [1,N_s]} \phi_i \\
\text{subject to} \quad & \textit{Relaxed } R \in SO(3), \\
& \phi_i \geq \mathbb{1}^T \alpha_i, \\
& \phi_i \geq \phi_{max}\, o_i, \\
& \alpha_i \geq + \left(Rs_i + T - MC_i^T\right) - \mathbb{M} o_i, \qquad (2) \\
& \alpha_i \geq - \left(Rs_i + T - MC_i^T\right) - \mathbb{M} o_i, \qquad (3) \\
& \sum_{j \in N_m} C_{i,j} + o_i = 1, \\
& \sum_{k \in N_f} f_{i,k} + o_i = 1, \\
& 0 \leq C_{i,j} \leq F_j\, f_i^T, \\
& \phi_i,\ \alpha_i \geq 0, \\
& f_{i,j},\ o_i \in \{0, 1\}.
\end{aligned}
$$

5.2 Handling Multiple Objects

Using similar machinery to that employed to correspond to outliers, we can extend our formulation further to support multiple objects. We can extend the formulation to simultaneously optimize over multiple rotations and translations $\{R^1, T^1\}, ..., \{R^{N_b}, T^{N_b}\}$ for N_b separate bodies. Given a map $B \in \{0,1\}^{N_b \times N_f}$, where the $(i,j)^{th}$ entry indicates if face j is a member of body i, we can replace constraints (2) and (3) with the disjunction

$$\begin{aligned} &\forall i \in N_m, k \in N_b : \\ &\qquad \alpha_i \geq + \left(R^k s_i + T^k - MC^T_{i,:}\right) - \mathbb{M}(1 - B_{k,:} f^T_{i,:}), \\ &\qquad \alpha_i \geq - \left(R^k s_i + T^k - MC^T_{i,:}\right) - \mathbb{M}(1 - B_{k,:} f^T_{i,:}). \end{aligned}$$

where the expression $\mathbb{M}(1 - B_{k,:} f^T_{i,:})$ deactivates the constraint if the current assignment f does not assign scene point i to a face on body k.

5.3 Using Other Pose Estimation Methods as a Heuristic

A benefit of optimizing directly over the fundamental problem addressed by a wide class of pose estimation methods is that we can take advantage of solutions generated by those other methods by consuming them as candidate feasible solutions. The branch and bound algorithm (and solvers that implement it) is able to asynchronously consume feasible solution guesses as nodes in the search tree. These new feasible solutions provide upper bounds on the global optimal cost, which are used to prune bad nodes. Because a significant amount of search time is spent finding better feasible solutions (as can be seen in the results in e.g. Figure 4), getting better feasible solutions from faster but less-consistent pose estimation methods can improve the runtime of the global optimization. This ability also means that this formulation can be used to post-process the output of methods that trade consistency for efficiency in order to guarantee stable results without completely discarding the efficiency of the original method.

Given the MIP formulation described above and a candidate pose $\{\hat{R}, \hat{T}\}$ generated by any method, one can extract C, f, ϕ, and α via closest-point queries against the mesh models. The value of $\hat{R}$ directly determines which binary variables should be active in the piecewise-linear relaxation of $R \in SO(3)$.

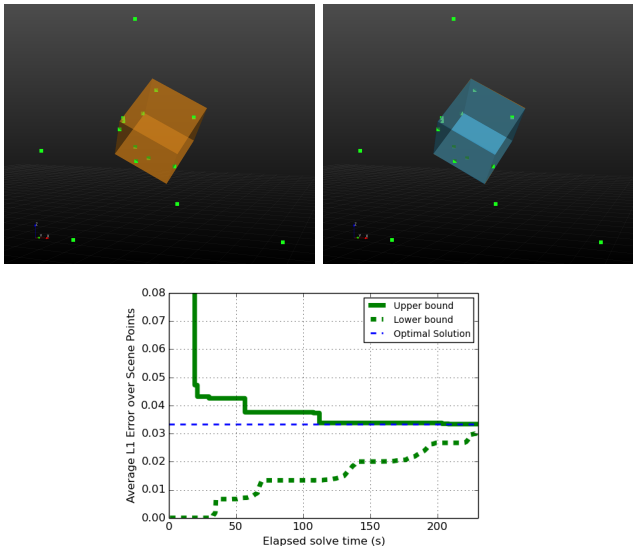


Fig. 4: Pose estimate produced by our MIP formulation with a cube model of 12 triangular faces, given 15 scene points with 5 outliers. The solution shown here has optimal cost that matches the optimal cost of the ground truth solution. Optimality of this solution was certified to a MIP gap of 5%. **Top left:** Ground truth pose. **Top right:** Pose estimate using our MIP formulation. **Bottom:** Convergence time of upper and lower bounds across time for the MIP solution.

6 Results

We executed several experiments to verify our formulation on synthetic data. In addition, we performed exploratory experiments on real data to probe the practicality of this and competing approaches to global pose estimation. To perform these experiments, we implemented both formulations in C++ and Julia, relying on the Drake [33] and JuMP [8] symbolic optimization libraries respectively. No performance difference between the libraries was observed. We used Gurobi 7.0.2 [15] as a backend to solve the resulting MIPs.

6.1 Experiments on Synthetic Data

6.1.1 General Performance

We generated a synthetic point cloud from a cube model with a side length of 1 unit. We generated 15 scene points, with 10 sampled randomly from the

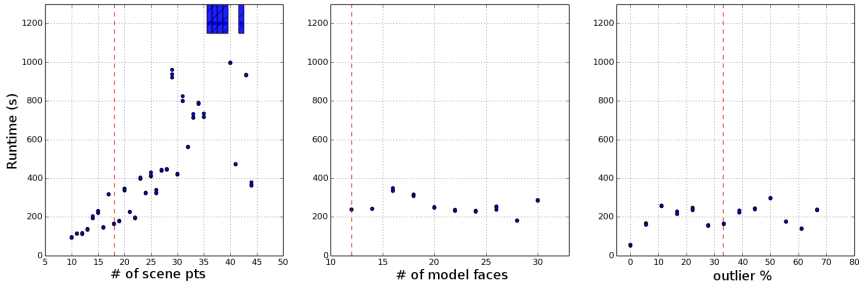


Fig. 5: The time scaling of the convergence time to a MIP gap of 5% as scene, model, and outlier complexity is varied. Each point represents a complete solve of a problem instance. Scene points were sampled uniformly from the surface of a cube described by varying numbers of triangular faces. The verticle red bar indicates the condition that is shared between each experiment (i.e. 18 scene points, 12 model faces, 33% outliers). **Left:** Solve times across a varying number of scene points, with 12 model faces and 33% intentionally injected outliers. Vertical blue bars indicate solutions that reached a maximum timeout of 1200s without converging to a MIP gap. **Middle:** Solve times across a varying number of model faces, with 18 scene points and 33% outliers. **Right:** Solve times across a varying set of outlier ratios, with 18 scene points and 12 model faces.

surface of the cube at its ground truth pose, and 5 more generated randomly in the area around the cube. We included outliers in this test case to illuminate how the solver performs in terms of progress of the upper *and lower* bounds – without outliers, the optimal error would be close to the trivial lower bound of 0. An optimal fit in this configuration has an optimal average saturated L-1 error of 0.033: the $\frac{2}{3}$ of points that are inliers have L-1 error of 0, and the $\frac{1}{3}$ of points that are outliers have L-1 error of $\geq \phi_{max} = 0.1$ by construction.

Our MIP formulation converged to the optimal solution and certified its global optimality to within a MIP gap of 5% (Figure 4). This desired MIP gap is tunable, and trades off with runtime. This gap of 5% was chosen arbitrarily, and corresponds to an optimality gap of $0.033 \times 0.05 = 0.00165$ average L-1 error over the scene points. We used 4 binary variables per element of R. The largest elementwise infeasibility of $R^T R = I$ being 0.020, and $det(R)$ was 1.002. These values reflect that the approximation of $R \in SO(3)$ is reasonably tight.

6.1.2 Characterization of Runtime

To characterize the time scaling of this algorithm with respect to model complexity, we recorded the runtime of the solver across a set of synthetic scene-model pairs with varying scene, model, and outlier complexities (Figure 5). Intriguingly, the characterization example had little noticeable runtime variation when model complexity was increased, even though the total problem size and number of binary variables scales linearly with the model complexity. It is possible that, since only the representation of the underlying geometry is changing (and not the underlying geometry itself), either the post-presolve MIP or its LP relaxations are not significantly different between these cases, in such a way to lead to similar runtimes. The case of zero outliers showed significantly lower runtime than the case of nonzero outliers, as the case of zero outliers has a trivial lower bound of 0 and thus an easier branch and bound search. In the case of nonzero outliers, however, there was no clear relationship between outlier ratio and runtime. These two properties make scaling this method to larger examples more hopeful, though it is unclear if the relationships will continue to hold for higher numbers of scene points and significantly more complex models.

6.1.3 Outlier Rejection and Multiple Models

To highlight the extensions of our formulation, we generated similar synthetic point clouds to test the outlier rejection and multiple model cases, with results shown in Figures 6 and 7. To avoid unreasonably long runtimes, we had to constrain rotations and limit the search to be over translations and correspondences. R was thus constrained to take the value of the ground truth rotation.

6.1.4 Upper Bounds from ICP

To demonstrate that solutions generated from other efficient but non-global methods can be leveraged to make our global optimization faster, we implemented an ICP-based heuristic for generating candidate feasible solutions online during the optimization. This procedure is directly inspired by GO-ICP [37]. Our solver maintains a queue of feasible solutions found by the branch and bound algorithm, and runs point-to-plane ICP with proportional outlier rejection on each feasible solution in a parallel thread alongside the global optimization solver. If the ICP produces a solution better than the best currently held by the solver, the ICP solution is handed to the solver as a heuristically-derived feasible solution. This procedure significantly improves runtime, as is shown in Figure 8.

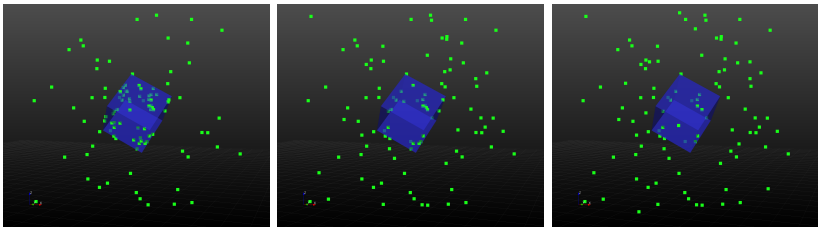


Fig. 6: Pose estimates produced by our MIP formulation for a cube model of 12 triangular faces, given 100 scene points, with a varying number of them being outliers: **Left:** 50% outliers, converged in 50s. **Middle:** 80% outliers, converged in 400s. **Right:** 90% outliers, converged in 2000s. Rotations were frozen to the ground truth rotation in order to produce these solutions in reasonable time. All solutions shown here have optimal cost that matches the optimal cost of the ground truth solution and align with the ground truth pose. Optimality of these solutions were certified to a MIP gap of 5%.

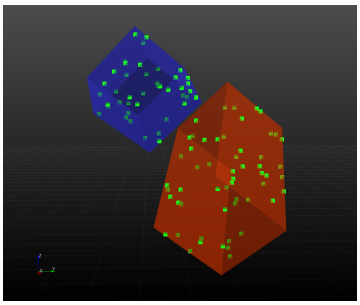


Fig. 7: Pose estimates produced by our MIP formulation simultaneously fitting two box models to 100 scene points with no outliers. Rotations were frozen to the ground truth rotations in order to produce these solutions in reasonable time. Convergence took 1100s.

6.2 Experiments on Real-World Data

To illustrate how this method compares to other global pose estimation methods when applied to real-world data, we provide experiments comparing the performance of GO-ICP [37], Super4PCS [24], and our method (MIP) on a set of examples drawn from point clouds collected from office and laboratory environments [21]. Examples were generated by cropping each complete point cloud (Figure 9) around the object of interest. No additional cleanup was performed to remove outliers or other clutter, and each method was supplied with only this raw cropped point cloud. GO-ICP and Super4PCS were chosen as representative methods from the approaches of branch-and-bound search and sampling methods respectively. The parameters of each method were hand-tuned to improve performance, but held constant across the entire dataset. In order to keep timings reasonable, we terminated our MIP solver after 60 seconds and took as its answer its current best feasible solution. We used ICP to provide candidate feasible solutions as described in Section 6.1.4, resulting in the MIP system functioning as a hybrid method that used ICP

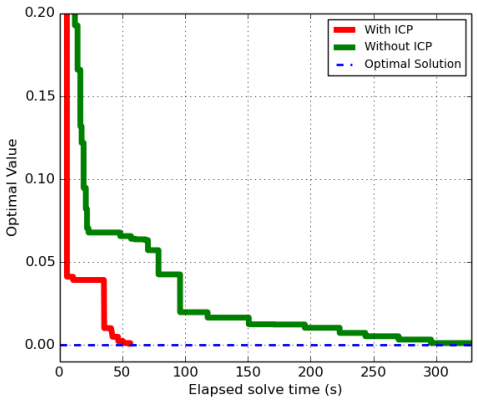


Fig. 8: Comparison of the upper bound convergence behavior of the MIP formulation with 50 scene points and 0 outliers fitting a box model, with and without an ICP algorithm generating novel feasible solutions in parallel. The lower bound is omitted, as it is trivially 0 for the 0 outlier case.

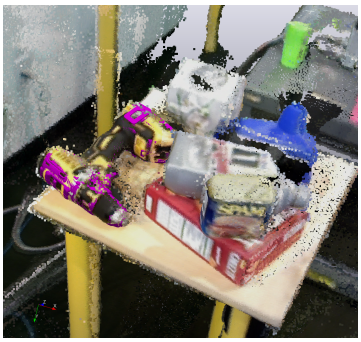


Fig. 9: An example pointcloud from our testing dataset. The ground-truth location of the object of interest in the pointcloud (in this case, the drill on the table) is known precisely. Test cases are drawn by cropping the point cloud around the known object location with varying crop sizes.

to make local progress, while using MIP heuristics and branch-and-bound search to generate new seeds for ICP to explore.

While each method was successful on a number of examples (e.g. Figure 10), this dataset proved extremely difficult for all methods (Figure 11). In almost all cases, incorrect solutions were the consequence of clutter and incorrect outlier handling.

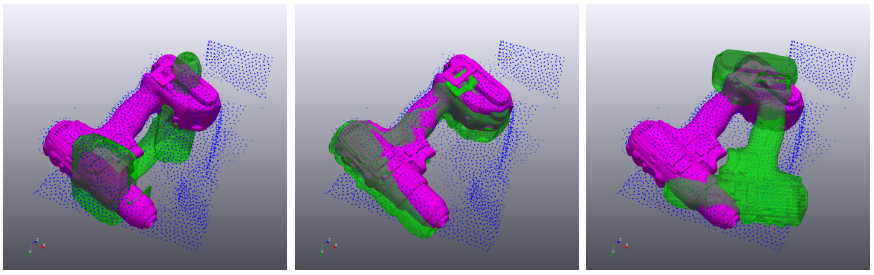
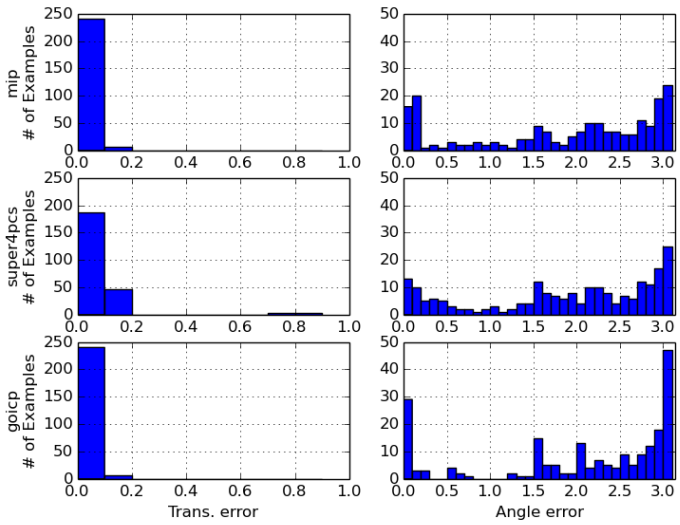


Fig. 10: Example performance of GOICP (**Left**), our solver's best solution after 60s (**Middle**), and Super4PCS (**Right**), given 25cm crops around the true object location without additional outlier removal. The ground-truth object location is shown in purple.

7 Conclusion

The formulations we present can be used to find certified ϵ-globally-optimal solutions for small numbers of scene points and outliers, even in the face of combinatorial complexity. The solver is capable of finding and certifying the right solution, even in very high outlier ratios, and can optimize with multiple objects seamlessly. That this technique can so easily incorporate hypotheses from other methods makes it a candidate for being an offline *verification* technique for the results from other efficient but inconsistent pose estimation methods. This functionality is critical when considering the kinds of highly ambiguous point clouds that result from highly cluttered scenes, and from tactile sensing. However, pilot experiments on real data from natural cluttered scenes results underline how elusive reliable global object pose estimation in complex, practical data remains. A truncated form of our method can hold its own in terms of performance against competing global pose estimation techniques, when compared on difficult, unsegmented, colorless point clouds. Further work is required to remove the early-stopped requirement and make it tractable to perform the branch and bound search to convergence for practically sized point clouds and models, as well as to incorporate RGB and other feature cues.

By framing the core point-cloud object pose estimation problem as a mixed-integer convex program, our solution method makes clear what it means to examine partial relaxations of the problem; and in doing so, makes it possible to verify that solutions are approximately globally optimal, or provide a search region that may contain better solutions. Further, the class of mixed integer convex programs seems sufficiently rich to capture many of the nuances of this problem – [6] shows that, in a similar mixed-integer convex framework with maximal coordinate rigid body configurations, it is possible to enforce complex kinematic constraints including revolute joints



	GOICP	MIP	Super4PCS
% correct	13.0%	12.6%	8.9%

Fig. 11: Histogram of translation and angle error of pose estimate of each method across a set of 246 test instances. The MIP results presented here were generated by taking the best solution found by the branch-and-bound search after 60s, and thus are not certified ϵ-optimal. While the object-centered cropping made the translation easy to estimate, the extreme clutter of the examples greatly injured all methods. Approximate rotational symmetries in the most common object (a drill) caused many erroneous solutions with 180-degree error, emphasizing the vulnerability of even these global methods to confusing clutter. Using a threshold of 5cm translation error and 10 degree angle error, only approximately 10% of examples were estimated correctly by a given method.

and nonpenetration with an environment. While object tracking algorithms (e.g. [30]) have had great success taking advantage of articulations, nonpenetration, and free-space information to combat ambiguity and refine their results, there remains a great opportunity for global pose estimation algorithms to utilize these additional sources of information. We hope that the tools and techniques we have presented will lead to more reliable and informed algorithms in these directions that can tackle object pose estimation in the complex and occluded scenes that are unavoidable as our robots move into the natural world.

Acknowledgements This material is based upon work supported by NSF Contract IIS-1427050, a National Science Foundation Graduate Research Fellowship under Grant No. 1122374, as well as support from ABB and Draper Laboratory.

References

1. C. B. Akgül, B. Sankur, Y. Yemez, and F. Schmitt. 3d model retrieval using probability density-based shape descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1117–1133, 2009.
2. M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri. Global motion estimation from point matches. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 81–88. IEEE, 2012.
3. P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.
4. K. N. Chaudhury, Y. Khoo, and A. Singer. Global registration of multiple point clouds using semidefinite programming. *SIAM Journal on Optimization*, 25(1):468–501, 2015.
5. Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992.
6. H. Dai, G. Izatt, and R. Tedrake. Global inverse kinematics via mixed-integer convex optimization. 2017.
7. B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 998–1005. Ieee, 2010.
8. I. Dunning, J. Huchette, and M. Lubin. Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.
9. D. W. Eggert, A. Lorusso, and R. B. Fisher. Estimating 3-d rigid body transformations: A comparison of four major algorithms. *Mach. Vision Appl.*, 9(5-6):272–290, Mar. 1997.
10. O. Enqvist, K. Josephson, and F. Kahl. Optimal correspondences from pairwise constraints. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1295–1302. IEEE, 2009.
11. M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
12. N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *Symposium on geometry processing*, volume 2, page 5, 2005.
13. R. I. Hartley and F. Kahl. Global optimization through rotation space search. *International Journal of Computer Vision*, 82(1):64–79, 2009.
14. S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian conference on computer vision*, pages 548–562. Springer, 2012.
15. G. O. Inc. Gurobi optimizer reference manual, 2016.
16. G. Izatt, G. Mirano, E. Adelson, and R. Tedrake. Tracking objects with point clouds from vision and touch. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017.
17. A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449, 1999.
18. M. Klingensmith, M. C. Koval, S. S. Srinivasa, N. S. Pollard, and M. Kaess. The manifold particle filter for state estimation on high-dimensional implicit manifolds. *arXiv preprint arXiv:1604.07224*, 2016.

19. H. Li and R. Hartley. The 3d-3d registration problem revisited. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.
20. S. Li, S. Lyu, and J. Trinkle. State estimation for dynamic systems with intermittent contact. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 3709–3715. IEEE, 2015.
21. P. Marion, P. R. Florence, L. Manuelli, and R. Tedrake. A pipeline for generating ground truth labels for real rgbd data of cluttered scenes. *Under Review*, 2017.
22. H. Maron, N. Dym, I. Kezurer, S. Kovalsky, and Y. Lipman. Point registration via efficient convex relaxation. *ACM Transactions on Graphics (TOG)*, 35(4):73, 2016.
23. G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part iconvex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.
24. N. Mellado, D. Aiger, and N. J. Mitra. Super 4pcs fast global pointcloud registration via smart indexing. In *Computer Graphics Forum*, volume 33, pages 205–215. Wiley Online Library, 2014.
25. V. Narayanan and M. Likhachev. Deliberative object pose estimation in clutter. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3125–3130. IEEE, 2017.
26. G. L. Nemhauser and L. A. Wolsey. Integer programming and combinatorial optimization. *Wiley, Chichester. GL Nemhauser, MWP Savelsbergh, GS Sigismondi (1992). Constraint Classification for Mixed Integer Programming Formulations. COAL Bulletin*, 20:8–12, 1988.
27. C. Olsson, F. Kahl, and M. Oskarsson. Branch-and-bound methods for euclidean registration problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):783–794, 2009.
28. C. Papazov and D. Burschka. An efficient ransac for 3d object recognition in noisy and occluded scenes. *Computer Vision–ACCV 2010*, pages 135–148, 2011.
29. D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard. Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group, 2016.
30. T. Schmidt, K. Hertkorn, R. Newcombe, Z. Marton, M. Suppa, and D. Fox. Depth-based tracking with physical constraints for robot manipulation. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 119–126. IEEE, 2015.
31. T. Schmidt, R. Newcombe, and D. Fox. Dart: dense articulated real-time tracking with consumer depth cameras. *Autonomous Robots*, 39(3):239–258, 2015.
32. J. Schulman, A. Lee, J. Ho, and P. Abbeel. Tracking deformable objects with point clouds. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1130–1137. IEEE, 2013.
33. R. Tedrake and the Drake Development Team. Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems, 2016.
34. F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *European Conference on Computer Vision*, pages 356–369. Springer, 2010.
35. P. Wohlhart and V. Lepetit. Learning descriptors for object recognition and 3d pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
36. J. M. Wong, V. Kee, T. Le, S. Wagner, G.-L. Mariottini, A. Schneider, L. Hamilton, R. Chipalkatty, M. Hebert, D. Johnson, et al. Segicp: Integrated deep semantic segmentation and pose estimation. *arXiv preprint arXiv:1703.01661*, 2017.
37. J. Yang, H. Li, D. Campbell, and Y. Jia. Go-icp: a globally optimal solution to 3d icp point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2241–2254, 2016.
38. Q.-Y. Zhou, J. Park, and V. Koltun. Fast global registration. In *European Conference on Computer Vision*, pages 766–782. Springer, 2016.