Stable Object Placement Planning From Contact Point Robustness

Philippe Nadeau and Jonathan Kelly, *Senior Member, IEEE*

***Abstract*—We introduce a planner designed to guide robot manipulators in stably placing objects within complex scenes. Our proposed method reverses the traditional approach to object placement: our planner selects contact points first and then determines a placement pose that solicits the selected points. This is instead of sampling poses, identifying contact points, and evaluating pose quality. Our algorithm facilitates stability-aware object placement planning, imposing no restrictions on object shape, convexity, or mass density homogeneity, while avoiding combinatorial computational complexity. Our proposed stability heuristic enables our planner to find a solution about 20 times faster when compared to the same algorithm not making use of the heuristic and eight times faster than a state-of-the-art method that uses the traditional sample-and-evaluate approach. The proposed planner is also more successful in finding stable placements than the five other benchmarked algorithms. Derived from first principles and validated in ten real robot experiments, our approach provides a general and scalable solution to the problem of rigid object placement planning.**

***Index Terms*—Assembly, manipulation planning, object placement, task planning.**

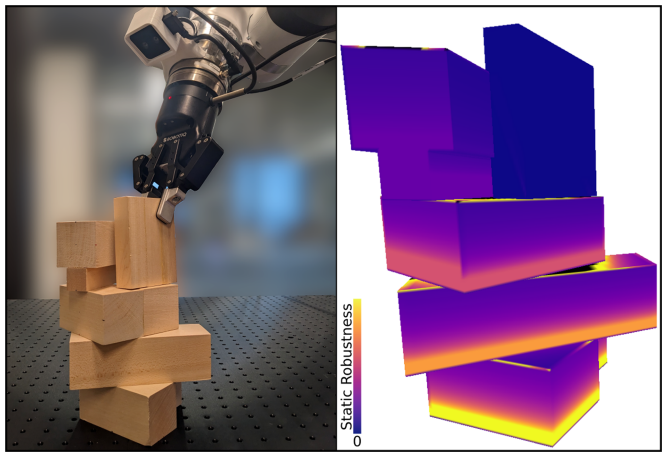


Fig. 1. Intricate, stable structure generated by our object placement planner. The surface is shaded according to robustness to perturbation by external forces; we use this measure to inform our proposed planner.

I. Introduction

ROBOTS with the capability to stably place objects in contact with one another hold the potential to reduce the need for human intervention in various tasks, spanning home chores, industrial operations, and work in outdoor environments [1]. In domestic settings, tasks like tidying a living space by rearranging objects or organizing garage storage could be automated [2]. Effective planning for stable object placement could facilitate safe palletizing of mixed products or enable the autonomous loading of trucks in industrial settings. In outdoor environments, autonomous excavators could be used in disaster relief through debris removal [3]. However, stably packing or rearranging rigid objects in contact is challenging due to the subtle yet influential force interactions between the objects that determine their stability. Since the problem of determining force interactions between rigid objects in frictional contact is known to be NP-hard [4], many approaches resort to planning heuristics based on shape information only and do not generalize well across tasks. Moreover, while existing methods have typically considered geometry and dynamics in two separate steps [5], [6] we propose to 1) combine them through a *static robustness map*, as shown in Fig. 1, and 2) define a planner that leverages this map to generate stable placements more efficiently—our two main contributions.

A general approach to placement planning requires considering the potential motion of objects under forceful interactions. In static assemblies, the mass and center of mass (henceforth referred to as the inertial parameters), as well as the friction coefficients, are sufficient to characterize an object's resistance to acceleration. A planner that operates based on these parameters is called *inertia-aware* in this work, with knowledge of the second mass moments not being required in static scenarios. In sum, this article introduces an inertia-aware object placement planning algorithm, leveraging a physically grounded heuristic to generate stable placements of known objects, with scalability to tackle large-scale problems.

The rest of this article is organized as follows. Section II reviews the literature on assembly stability and object placement planning, highlighting the novelty of our proposed algorithm. In Section III, the inertia-aware object placement planning problem is formally defined, while Section IV introduces a physically grounded heuristic to assess an assembly's capacity to

Received 6 December 2024; revised 16 April 2025; accepted 21 May 2025. Date of publication 6 June 2025; date of current version 23 June 2025. This work was supported in part by the Canada Research Chairs program. This article was recommended for publication by Associate Editor W. Wan and Editor J. Bohg upon evaluation of the reviewers' comments. *(Corresponding author: Philippe Nadeau.)*

The authors are with the STARS Laboratory, Institute for Aerospace Studies, University of Toronto, Toronto, ON M3H 5T6, Canada (e-mail: philippe.nadeau@robotics.utias.utoronto.ca; jonathan.kelly@robotics.utias.utoronto.ca).

Jonathan Kelly is a Vector Institute Faculty Affiliate.

This article has supplementary downloadable material available at https://doi.org/10.1109/TRO.2025.3577049, provided by the authors.

Digital Object Identifier 10.1109/TRO.2025.3577049

withstand external forces. Our proposed planner is detailed in Section V, and the results of more than 1500 simulation experiments involving six scenes and six algorithms are presented in Section VI, demonstrating the benefits of our method. In Section VII, practical validation of our proposed algorithm is carried out through 10 real-world experiments involving 50 placements, in which a robot manipulator builds assemblies while recording any failures that occur. Finally, Section VIII concludes this article.

II. Related Work

Early methods to produce stable orientations of an assembly under gravity, for fixturing purposes, include [7], which ignores friction. In [8], friction is taken into account but stability is not guaranteed due to indeterminacies in the distribution of contact forces [9], a common issue. Selecting a set of feasible contact forces can be done by resorting to optimization-based methods [10], [11] that make use of the principle of virtual work [12], and integrating kinematic constraints to ensure that the solution is physically plausible. Such methods have been shown to be equivalent to finite element methods (FEM) [13] for rigid objects, and, similar to FEM, are computationally expensive and nondeterministic. Stability assessment under a large number of random force disturbances is perfomed in [5] by solving a *min–max* optimization problem for every perturbation, making it the primary bottleneck of the planning algorithm. We instead propose a physically sound heuristic that is quick to compute by avoiding combinatorial complexity. Also, in contrast to [5], our approach considers stability first instead of verifying it only at the end.

Classical methods used to assess the stability of a grasp [14], [15] can provide a quantitative evaluation of the stability of a workpiece, while newer algorithms can produce stable multi-object grasps [16] with rigid convex objects. Although these methods offer a comparative basis for different placements, their application to object placement planning is limited. This is because they do not take into account the fact that fixed rigid objects can withstand very large forces, nor do they guide the planner on how to position an object on a stable structure– - two aspects that our proposed algorithm addresses.

In general, the assembly task planning problem has been shown to be NP-complete [17], which explains why task and motion planning methods [18] have only been applied to small-scale problem instances. Learning-based methods that do not resort to handcrafted heuristics have been proposed for assembly planning [19], [20] but also struggle to generalize across tasks [21]. While most object planning algorithms ignore the inertial parameters of the objects, the work in [6] make use of the center of mass of the object being placed but is limited to isolated placements on fixed horizontal surfaces. Through our experiments, we show examples where this common approach may produce unstable placements. As an attempt at a more general object placement planning algorithm, physics simulators have been integrated into planning algorithms [22] but are prohibitively slow to generate valid plans, even for small-scale problems. In contrast, this work proposes a general method derived from first principles that avoids the overhead of dynamics simulators to better scale to larger problems.

Planning stable placements not only requires determining reaction forces, but also the set of contact points forming the interfaces between objects, which is a complex task in general. Hence, previous works have simplified the problem by limiting the shape of the objects to rectangular workpieces [5], [23], or by segmenting the scene into simple shapes like cylinders and cuboids [24]. In contrast, our method can handle any shape described by a triangular mesh, with the placement planning time growing sub-quadratically with the number of vertices.

In this work, we tackle the planning aspects of the inertia-aware object placement planning problem. However, a complete system would also include a perception pipeline to identify the properties of unknown objects. Particularly relevant to placement planning in static conditions, the shape and inertial parameters of a manipulated object are accurately identified in [25] with a commonly available force–torque sensor. Moreover, the approach in [26] uses tactile sensors to estimate the inertial parameters and the friction of manipulated objects, which could enable our method to deal with unknown objects.

III. Inertia-Aware Object Placement Planning

Let $\mathcal{O}$ be a set of N objects with known shape $\mathcal{V}_i \subset \mathbb{R}^3$ and material composition $\rho_i(\mathbf{p})\ \forall\ \mathbf{p} \in \mathcal{V}_i$ for $i \in \{1, \ldots, N\}$. Let an assembly $\mathcal{O}_a \subset \mathcal{O}$ be a set of objects in frictional contact, and $\mathcal{O}_f \subset \mathcal{O}_a$ be the subset of objects fixed in the limited space of the scene. The problem is to find a sequence of penetration-free placement poses $\mathbf{T} \in \mathrm{SE}(3)$ that maximizes the force needed to displace any object. Hence, different from typical assembly planning problems, the goal configuration is not given in terms of the pose of the objects in the assembly, but rather in terms of an objective that is to be maximized.

Unless stated otherwise, the RIGID notation convention [27] is used and ${}^c_b\mathbf{p}_a$ is the position vector of $\{a\}$ with respect to $\{b\}$ and expressed in $\{c\}$. The orientation of $\{a\}$ relative to $\{b\}$ is given by ${}_b\mathbf{R}_a$, and a unit-length vector $\mathbf{v}$ is denoted by $\hat{\mathbf{v}}$. The skew-symmetric operator $[\cdot]_\times$ is defined such that $[\mathbf{u}]_\times \mathbf{v} = \mathbf{u} \times \mathbf{v}$ for the vectors $\mathbf{u}$ and $\mathbf{v}$ in $\mathbb{R}^3$. An identity matrix of size $n \times n$ is denoted $\mathbf{1}_n$.

IV. Static Robustness Assessment

When placing an object amongst others, it is usually desirable that the object be unlikely to move after being placed. For rigid polyhedral objects, ultimately, instability will occur when an object slips or topples. In both cases, the event is triggered when an applied force exceeds a threshold that we call the *static robustness:* the maximum amount of force that can be exerted along a given direction, on a given point, before an object in the assembly moves. Formally, the static robustness r is defined as

$$r : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}_+ \tag{1}$$

that maps a position $\mathbf{p} \in \mathbb{R}^3$ and direction $\hat{\mathbf{e}} \in \mathbb{S}^2$ to a scalar value r. It follows that planning for stable placement involves reasoning about contact forces in the assembly. However, computing frictional contact forces in an assembly has been shown

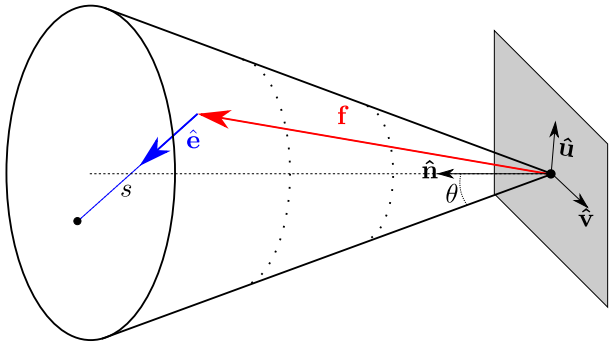


Fig. 2. Friction cone at a contact point with reaction force $\mathbf{f}$ (red) and an external force $\hat{\mathbf{e}}$ (blue). Increasing the magnitude of the external force vector to s will produce an intersection with the boundary of the cone at $\mathbf{f} + s\,\hat{\mathbf{e}}$.

to be NP-hard [4], and is made more complex by the existence of a coupling between forces and kinematic constraints [28]. In this work, we make the following assumptions about object dynamics.

1) The Coulomb friction model is in effect.
2) Objects are very hard and only deform at contact points.
3) Contact points deform under the law of linear elasticity.

With the two last assumptions being common with the Hertzian contact model [29].

A. Solving for Reaction Forces

In order to be quick to compute, a successful heuristic will need to avoid any computation whose complexity grows exponentially with the number of objects in the assembly. Hence, we solve a linearly constrained quadratic program [10] that uses a pyramidal approximation of the friction cone, enforce static equilibrium, and avoid tensile forces. This convex optimization problem can be solved in polynomial time [30] by standard solvers [31], but may find unphysical solutions due to the approximation of the friction cone. Focusing on a single object with I contact points, let

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \cdots & \mathbf{A}_I \end{bmatrix} \text{ with } \mathbf{A}_i = \begin{bmatrix} \mathbf{1}_3 \\ [{}_w\mathbf{p}_i]_\times \end{bmatrix} \boldsymbol{\mathcal{F}}_i \tag{2}$$

be a data matrix where ${}_w\mathbf{p}_i$ is the position of the ith contact point on the object and $\boldsymbol{\mathcal{F}}_i = \begin{bmatrix} \hat{\mathbf{u}}_i & \hat{\mathbf{v}}_i & \hat{\mathbf{n}}_i \end{bmatrix}$ defines a local frame with $\hat{\mathbf{n}}_i$ being the inward normal to the contact surface as pictured in Fig. 2. Let

$$\mathbf{b} = -m \begin{bmatrix} \mathbf{1}_3 \\ [{}_w\mathbf{p}_c]_\times \end{bmatrix} \mathbf{g} \tag{3}$$

be the wrench exerted by gravity on the center of mass ${}_w\mathbf{p}_c$ of the object whose mass is m. The system of equations involving the forces $\mathbf{f} = \begin{bmatrix} {}_w\mathbf{f}_1^\mathsf{T}, \ldots, {}_w\mathbf{f}_I^\mathsf{T} \end{bmatrix}^\mathsf{T}$ on the object is

$$\mathbf{A}\mathbf{f} = \mathbf{b} \tag{4}$$

where ${}_w\mathbf{f}_i$ is the reaction force exerted at the ith contact point and measured in the world frame. The reaction force at the ith contact point expressed in $\boldsymbol{\mathcal{F}}_i$ is ${}^i_w\mathbf{f}_i$ such that components ${}^i_w\mathbf{f}_{i_n}$ and ${}^i_w\mathbf{f}_{i_t}$ are, respectively, perpendicular and parallel to the contact surface. When dealing with multiple objects, $\mathbf{A}$ and $\mathbf{b}$ are constructed for every object and concatenated vertically. Note that, as per D'Alembert's principle, a dynamical system can be reduced to an equivalent static system by considering inertial forces that are due to the motion of the system. Including all inertial forces in (3) would enable our method to handle dynamic scenarios as well.

The principle of virtual work [12] dictates that the true distribution of forces should minimize the sum of squared forces for rigid objects in static equilibrium, such that

$$\begin{aligned} \min_{\mathbf{f} \in \mathbb{R}^{3n}} \quad & \mathbf{f}^\mathsf{T}\mathbf{f}/2 \\ \text{s.t.} \quad & \mathbf{A}\mathbf{f} = \mathbf{b} \\ & \left\| {}^i_w\mathbf{f}_{i_n} \right\|_2 \geq 0 \qquad \forall i \in \{1, \ldots, I\} \\ & \left\| {}^i_w\mathbf{f}_{i_t} \right\|_1 \leq \mu_i \left\| {}^i_w\mathbf{f}_{i_n} \right\|_2 \qquad \forall i \in \{1, \ldots, I\} \end{aligned} \tag{5}$$

determines reaction forces with μ_i being the friction coefficient at the ith contact point. The optimization problem in (5) is a linearly constrained quadratic program (QP), for which the optimal solution can be obtained with standard solvers [31] if it exists. The constraints in (5), respectively, enforce the equilibrium of the assembly, the compressive nature of the contact forces, and a pyramidal approximation of the Coulomb friction cone.

Alternatively, the minimal two-norm solution can be obtained via a QR decomposition [32] of the data matrix, with

$$\mathbf{Q}, \mathbf{R} = \text{qr}\left(\mathbf{A}^\mathsf{T}\right) \tag{6}$$

by solving

$$\mathbf{R}^\mathsf{T}_{1:6,1:6}\mathbf{z} = \mathbf{b} \tag{7}$$

for $\mathbf{z}$, where $\mathbf{R}_{1:6,1:6}$ is the upper-left 6×6 sub-matrix of $\mathbf{R}$. The reaction forces can then be obtained with

$$\mathbf{f} = \mathbf{Q}_{:,1:6}\mathbf{z} \tag{8}$$

where $\mathbf{Q}_{:,1:6}$ is built from the first six columns of $\mathbf{Q}$. This approach is expected to be the fastest but is unconstrained and might violate the friction cone and the nontensile force constraints. For placement planning, the QR-based approach can be used during the iterative process by checking the magnitudes of the constraint violations and verifying the validity of promising solutions with the optimization-based approach. However, due to static indeterminacies, the existence of a solution to the optimization problem in (5) does not guarantee that the assembly is stable [9]. Stability can only be guaranteed by testing all possible solutions. If needed, a different method could be used to compute reaction forces, while still seamlessly integrating with our proposed planner.

B. Robustness to Slipping

Consider the friction cone at a given contact point, defined by its local frame $\boldsymbol{\mathcal{F}}_i = \begin{bmatrix} \hat{\mathbf{u}}_i & \hat{\mathbf{v}}_i & \hat{\mathbf{n}}_i \end{bmatrix}$ and angle θ, as shown in Fig. 2. Let $\mathbf{f}$ and $\hat{\mathbf{e}}$ be a reaction force vector and a unit external force vector at the contact point, respectively. The maximal magnitude of the external force vector $\mathbf{e} = s\,\hat{\mathbf{e}}$ that

can be applied to the contact point is the one that produces an intersection between the friction cone and the line defined by the external force vector.

The general equation defining a circular cone embedded in 3-D space and supported by angle θ is given by

$$\tan^2(\theta)z^2 = x^2 + y^2 \tag{9}$$

where $\tan^2\theta = \mu^2$, the square of the friction coefficient at the contact point. Let ${}^{i}_{w}\hat{\mathbf{e}}_i = \begin{bmatrix} e_u & e_v & e_n \end{bmatrix}^{\mathsf{T}}$ and ${}^{i}_{w}\mathbf{f}_i = \begin{bmatrix} r_u & r_v & r_n \end{bmatrix}^{\mathsf{T}}$ be the components of the forces expressed in $\boldsymbol{\mathcal{F}}_i$, and s be the magnitude of the external force vector. A point on the cone boundary can be given as a function of the forces at the contact point with

$$x = se_u + r_u \tag{10}$$

$$y = se_v + r_v \tag{11}$$

$$z = se_n + r_n \tag{12}$$

as shown in Fig. 2. Substituting (10) into (9) yields

$$\tan^2(\theta)(se_n + r_n)^2 = (se_u + r_u)^2 + (se_v + r_v)^2 \tag{13}$$

which is a quadratic in s

$$0 = as^2 + bs + c \tag{14}$$

where

$$a = \mu^2 e_n^2 - e_u^2 - e_v^2 \tag{15}$$

$$b = 2\left(\mu^2 r_n e_n - r_u e_u - r_v e_v\right) \tag{16}$$

$$c = \mu^2 r_n^2 - r_u^2 - r_v^2 \tag{17}$$

and for which a particular solution is given by

$$s = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \tag{18}$$

the well-known quadratic formula.

Hence, the static robustness r of a contact point is given by

$$r = \begin{cases} s & \text{if } s > 0 \\ \infty & \text{otherwise.} \end{cases} \tag{19}$$

The static robustness is infinite if $\hat{\mathbf{e}} + \mathbf{f}$ does not intersects with the friction cone. Since the reaction forces are not constrained to obey the nonlinear Coulomb friction model in the computation of (8), it is possible that

$$\mu \left\| {}^{i}_{w}\mathbf{f}_{i_n} \right\| < \left\| {}^{i}_{w}\mathbf{f}_{i_t} \right\| \tag{20}$$

$$\mu^2 r_n^2 < \left(r_u^2 + r_v^2\right) \tag{21}$$

$$c < 0 \tag{22}$$

which implies that the contact point is not holding. In such a case, or similarly if the radicand $b^2 - 4ac$ is negative, the static robustness of the contact point is set to zero. Assuming that contact points obey the laws of linear elasticity, the stress incurred at each contact point is cumulated to produce the total frictional force [33], [34]. Hence, the robustnesses of several contact points are summed with

$$r_{\text{slip}} = \sum r_i \quad \forall i \in \{1, \ldots, I\} \tag{23}$$

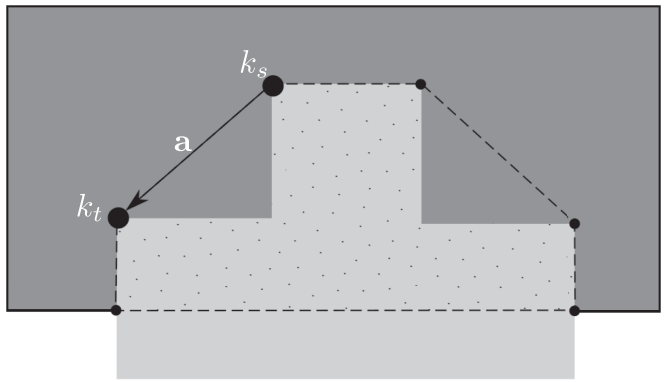


Fig. 3. View from above of a nonconvex object (light gray) atop another object (dark gray). The edges of the convex hull (dashed line) of the contact points are potential toppling axes.

to produce the robustness of the object to slipping, where r_i is the robustness of the ith contact point.

C. Robustness to Toppling

The robustness of an object to being pushed over is given by the minimum amount of force required to rotate the object about some axis $\mathbf{a}$, defined by the line joining a pair of contact points as shown in Fig. 3. For an isolated object being pushed, the toppling axis is guaranteed to be an edge of the convex hull of the contact points. Any other axis would require a greater amount of force to rotate the object because the lever arm would be smaller. Hence, the convex hull of the n contact points is computed via the QuickHull algorithm [35], whose complexity is $O(n^2)$ in the worst case, to produce a list of axes about which the object could topple.

Let, ${}^{w}_{k_s}\mathbf{p}_c$ be the location of the center of mass relative to k_s and expressed in the world frame, and $m\mathbf{g}$ be the weight of the object. An axis is a *valid* (i.e., possible) toppling axis if each contact point normal exerts a positive torque about the axis, which is verified with

$$\left({}^{w}_{k_s}\mathbf{p}_i \times {}_{w}\mathbf{f}_{i_n}\right) \cdot \mathbf{a} > 0 \quad \forall i \in \{1, \ldots, I\} \tag{24}$$

and that the weight of the object exerts a negative torque about the axis, which is verified with

$$\left({}^{w}_{k_s}\mathbf{p}_c \times m\mathbf{g}\right) \cdot \mathbf{a} < 0. \tag{25}$$

The robustness of an object to toppling about $\mathbf{a}$ is given by

$$s_i = \frac{\left({}^{w}_{k_s}\mathbf{p}_c \times m\mathbf{g}\right) \cdot \mathbf{a}}{\left({}^{w}_{k_s}\mathbf{p}_i \times \hat{\mathbf{e}}\right) \cdot \mathbf{a}} \tag{26}$$

which is the force required to compensate for the torque due to gravity that is holding the object in place. The maximum amount of force that can be exerted along a given direction on a given point before the object topples is given by

$$r_{\text{top}} = \min\{s_i\} \quad \forall i \in \{1, \ldots, I\} \tag{27}$$

which provides the minimum toppling robustness when all valid axes are considered.

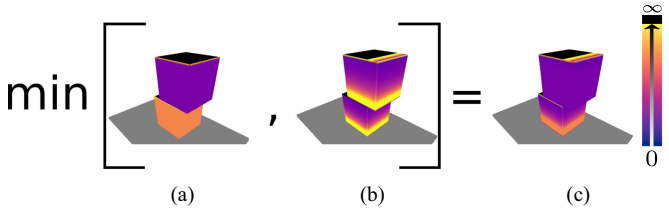


Fig. 4. Combining slipping (a) and toppling (b) robustnesses to obtain the overall static robustness (c) with values color-coded according to the scale on the right (black being infinite).

Slipping and toppling robustnesses may be combined as

$$r = \min\{r_{\text{slip}}, r_{\text{top}}\} \tag{28}$$

to obtain the overall static robustness of the object, as pictured in Fig. 4. With (28), the definition of the static robustness r as a mapping from a direction $\hat{\mathbf{e}}$ and position ${}_w\mathbf{p}_p$ to a scalar value r is fulfilled. The result of the mapping can be seen in Figs. 1, 4, and 9, where the color indicates the relative magnitude of the maximal force that can be applied on a point in the direction normal to the surface. A lighter color indicates that a greater force can be applied before the object slips or topples, with black indicating that an infinite force can be applied. The robustness function obtained via the use of our heuristic is unlikely to be exact in multiobjects assemblies. However, it is quick to compute because it avoids any operation whose computational complexity grows exponentially with the number of objects in the assembly.

V. Placement Planning From Static Robustness

Ideally, the normal contact force at each contact point is much greater than the tangential force such that the object is far from slipping. Therefore, our proposed planner uses the robustness to normal forces at surface points to plan stable placements. In the following, we assume that a known object has to be placed in a given assembly of objects for which the robustness in the normal direction at all points on the surface of the objects in the assembly is known. Also, placements are assumed to be performed in a linear (i.e., a single object is moved at a time), monotone (i.e., placing does not involve rearranging), and sequential (i.e., a single gripper is used) fashion [36].

By placing an object in an assembly, at least one contact interface between objects is created, which fixes some of the degrees of freedom (DoFs) of the object. Any nondegenerate[1] contact interface can be categorized into [37], [38] the following.

1) A face/face plane contact that fixes 3 DoFs.
2) A face/edge line contact that fixes 2 DoFs.
3) An edge/edge point contact that fixes 1 DoF.
4) A face/vertex point contact that fixes 1 DoF.

At least three contact interfaces are necessary to constrain the object to a specific pose by fixing all 6 DoFs [38]. However, to place an object successfully without inducing a collision, at least one DoF must remain unconstrained. With two noncollinear contact interfaces, from one (e.g., with face/face interfaces) to four (e.g., with face/vertex interfaces) DoFs will remain unconstrained, ensuring that the object can be placed without inducing a collision. A third contact interface might be needed to ensure stability under gravity, but the interface does not necessarily need to be selected prior to the computation of the placement pose. Furthermore, in a face/face contact situation, the normal vectors on each side of the contact interface must be opposed. In a face/edge contact, the normal vector of the face must be orthogonal to the contact edge.

[1]In [37], edge/vertex, vertex/vertex, and parallel edge/edge contacts are defined as being degenerate.

Algorithm 1: Placement Planning From SR.

input : Assembly $\mathcal{O}_a$, object to place $\mathcal{O}_i$, static robustness map $\mathrm{r}(\mathbf{p}_i, \hat{\mathbf{n}}_i) \ \ \forall\ \mathbf{p}_i \in \mathcal{P}$
output: Stable pose ${}_w\mathbf{T}_o$ of $\mathcal{O}_i$ if found

```
while max. number of iterations not reached do
    Sample two points ₙpₐ and ₙp_b on O_a with (29)
    Find two points ₒp_q and ₒp_r on the support
      geometry of O_i whose normals oppose those on
      the assembly and whose relative positions
      coincide
    Define ₙT_o from ₙpₐ, ₙp_b, ₒp_q, and ₒp_r
    if ₙT_o does not result in inter-penetration then
        Solve (8) for reaction forces (QR-based)
        if tension forces are below threshold then
            Solve (5) for reaction forces (QP-based)
            if forces are equilibrated then
                return ₙT_o
```

Our approach is summarized by Algorithm 1, with each step described in greater detail in the following sections. Our algorithm is based on sampling points on the assembly and on the object to place, and then defining a pose for the object based on the two sets of points. With this sampling-based approach, it is expected that many candidates will turn out to be invalid, and that the process will need to be iterated several times before a stable, nonpenetrating pose is found. Each iteration therefore needs to be performed as quickly as possible.

A. Contact Point Selection

The selection of candidate contact points on the surface of objects in the assembly is performed in an iterative, probabilistic manner based on the robustness of the objects to normal forces. At each iteration of the planning algorithm, two points are sampled according to the probability distribution in (29). With $\mathrm{r}(\mathbf{p}, \hat{\mathbf{n}})$ being the robustness of the object in the normal direction at point $\mathbf{p}$, the probability of sampling this point is given by

$$\mathrm{P}(\mathbf{p}) = \frac{\min(\mathrm{r}(\mathbf{p}, \hat{\mathbf{n}}), Q)}{\sum_{\mathbf{p}_i \in \mathcal{P}} \min(\mathrm{r}(\mathbf{p}_i, \hat{\mathbf{n}}_i), Q)} \tag{29}$$

where Q is the robustness above which the odds of being sampled are set to be constant. In practice, Q can be defined such that the largest finite robustness in the assembly has a small probability of being sampled (e.g., 10%). This can be done by initializing Q

to a larger value $Q_0 > 0$ that is exponentially decayed with the number of iterations k at a rate $\lambda < 1$ with $Q_k = Q_0 \lambda^k$. This way, the probability of sampling any point in $\mathcal{P}$ converges to

$$\lim_{k\to\infty} \mathrm{P}(\mathbf{p}) = \frac{Q_k}{|\mathcal{P}|Q_k} = \frac{1}{|\mathcal{P}|} \tag{30}$$

where $|\mathcal{P}|$ is the cardinality of $\mathcal{P}$, the set of surface contact points. With such a scheme, the odds of sampling a point with a large robustness are initially higher, but decrease with the number of iterations. All points are eventually considered with almost equal probability, and any valid placement pose is eventually considered.

1) Extension: Sampling on Fixed Supports: Sometimes, it might be desirable to place the object on a fixed support (e.g., table) instead of among the objects in the assembly. For instance, if placing the object on the assembly would result in a weakened assembly, it might be preferable to place the object on a fixed support whose static robustness is, by definition, infinite. In this case, however, it is usually desired that the object be placed in a compact manner. The probability function in (29) can be easily extended to accommodate this requirement. For instance, the set of scene points $\mathcal{P}$ can be augmented with a set of points $\mathcal{P}_f$ uniformly distributed on the fixed support with

$$w(\mathbf{p}) = \begin{cases} P(\mathbf{p}) & \forall \mathbf{p} \in \mathcal{P} \\ \max_{\mathbf{p}\in\mathcal{P}}(\mathrm{P}(\mathbf{p})) & \forall \mathbf{p} \in \mathcal{P}_f \end{cases} \tag{31}$$

$$w(\mathbf{p}) = w(\mathbf{p})\, e^{-\frac{\gamma^k}{s^2}\|{}_w\mathbf{p}_p - {}_w\mathbf{p}_c\|^2} \ \forall \mathbf{p} \in \mathcal{P}\cup\mathcal{P}_f \tag{32}$$

$$P(\mathbf{p}) = \frac{w(\mathbf{p})}{\sum w(\mathbf{p})} \ \forall \mathbf{p} \in \mathcal{P}\cup\mathcal{P}_f \tag{33}$$

where (32) increases the probability of sampling points closer to the centroid ${}_w\mathbf{p}_c$ of the scene and (33) ensures that the sum of the probabilities is equal to one. In (32), γ is the rate at which the probability of sampling a point close to the centroid of the scene decreases and s is the scale of the scene.

B. Object Point Selection

To define a placement pose, a pair of object features (i.e., faces or edges) is considered. One point per feature is selected such that the distance between the object points corresponds to the distance between the scene contact points. To favor stability, the object points are selected such that the line joining the points is as close as possible to the center of mass of the object. Prior to point selection, candidate pairs of features on the object are validated by considering the relative orientation of the feature normals to ensure that they can *afford* the support geometry of the contact points in the assembly. Each feature pair can either be *face-face* (parallel, coplanar, or intersecting), *face-edge* (parallel, or intersecting), or *edge-edge* (parallel, intersecting, or skew).

In the following, we assume that the contact points ${}_w\mathbf{p}_a$ and ${}_w\mathbf{p}_b$ are sampled on object faces in the scene with normals ${}_w\hat{\mathbf{n}}_a$ and ${}_w\hat{\mathbf{n}}_b$ such that $\|{}_w\mathbf{p}_a - {}_w\mathbf{p}_b\| = L$. The relative orientation of ${}_w\hat{\mathbf{n}}_a$ and ${}_w\hat{\mathbf{n}}_b$ can be obtained with

$$\hat{\boldsymbol{\omega}} = {}_w\hat{\mathbf{n}}_a \times {}_w\hat{\mathbf{n}}_b \tag{34}$$

$$\theta = \arccos({}_w\hat{\mathbf{n}}_a \cdot {}_w\hat{\mathbf{n}}_b) \tag{35}$$

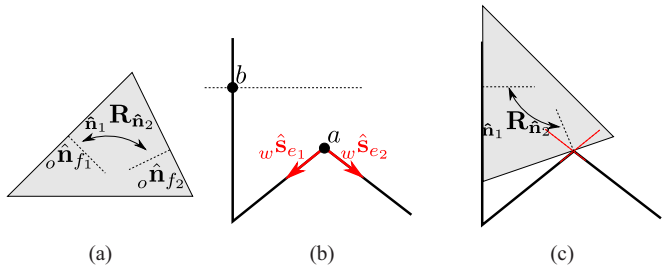


Fig. 5. Affordance test to see if ${}_o\hat{\mathbf{n}}_{f_2}$ would lie between ${}_w\hat{\mathbf{s}}_{e_1}$ and ${}_w\hat{\mathbf{s}}_{e_2}$ when ${}_o\hat{\mathbf{n}}_{f_1}$ opposes ${}_w\hat{\mathbf{n}}_b$.

$${}_a\mathbf{R}_b = \mathbf{1}_3 + \sin(\theta)[\hat{\boldsymbol{\omega}}]_\times + (1-\cos(\theta))[\hat{\boldsymbol{\omega}}]_\times^2 \tag{36}$$

where $\hat{\boldsymbol{\omega}}$ is the axis of rotation from ${}_w\hat{\mathbf{n}}_a$ to ${}_w\hat{\mathbf{n}}_b$, θ is the angle of rotation about $\hat{\boldsymbol{\omega}}$ from ${}_w\hat{\mathbf{n}}_a$ to ${}_w\hat{\mathbf{n}}_b$, and ${}_a\mathbf{R}_b$ is the rotation matrix from ${}_w\hat{\mathbf{n}}_a$ to ${}_w\hat{\mathbf{n}}_b$.

Validating Feature Pairs: A face will afford another face if its normal is parallel and opposed to the normal of the other face. In 3-D, an edge ${}_o\mathbf{g}_i$ joins two adjacent faces whose normals are denoted by ${}_o\hat{\mathbf{n}}_{e_1}$ and ${}_o\hat{\mathbf{n}}_{e_2}$. The vectors directed from the edge to the inside of the face, along the face and orthogonal to the edge are given by

$${}_o\hat{\mathbf{s}}_{e_1} = ({}_o\mathbf{g}_i \times {}_o\hat{\mathbf{n}}_{e_1}) / \|{}_o\mathbf{g}_i \times {}_o\hat{\mathbf{n}}_{e_1}\| \tag{37}$$

$${}_o\hat{\mathbf{s}}_{e_2} = ({}_o\mathbf{g}_i \times {}_o\hat{\mathbf{n}}_{e_2}) / \|{}_o\mathbf{g}_i \times {}_o\hat{\mathbf{n}}_{e_2}\| \tag{38}$$

and termed the *side vectors* [see Fig. 5(b)]. The object normals and side vectors are computed in the object frame, and the pose of the objects in the scene is used to project these vectors in the world frame, accommodating dynamically changing scenes.

A face with outward normal ${}_o\hat{\mathbf{n}}$ will afford an edge if the normal is orthogonal to the edge direction ${}_o\mathbf{u}_e$ such that

$${}_o\mathbf{u}_e \cdot {}_o\hat{\mathbf{n}} = 0 \tag{39}$$

and if the normal lies between the two side vectors of the edge. Intuitively, a planar surface with infinitesimal area should touch the edge without touching the faces on either side. A simple test to check if the normal lies between ${}_o\hat{\mathbf{s}}_{e_1}$ and ${}_o\hat{\mathbf{s}}_{e_2}$ is defined with

$$u = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} ({}_o\hat{\mathbf{s}}_{e_1} \times {}_o\hat{\mathbf{s}}_{e_2}) \tag{40}$$

$$v = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} ({}_o\hat{\mathbf{n}} \times {}_o\hat{\mathbf{s}}_{e_2}) \tag{41}$$

$$w = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} ({}_o\hat{\mathbf{n}} \times {}_o\hat{\mathbf{s}}_{e_1}) \tag{42}$$

such that the normal lies between the two side vectors if $uv \leq 0$ and $uw \geq 0$.

Since the normals sampled in the scene are expressed in a different frame than the ones on the object, the components of the normal vectors cannot be simply compared. Instead, the relative orientation of the object normals is considered to check if the scene can afford the object normals. For instance, assuming that the normal to the first feature selected on the object opposes ${}_w\hat{\mathbf{n}}_b$, the normal of the second feature needs to lie between the side vectors around point ${}_w\mathbf{p}_a$ when ${}_o\hat{\mathbf{n}}_{f_1}$ opposes ${}_w\hat{\mathbf{n}}_b$, as shown in Fig. 5.

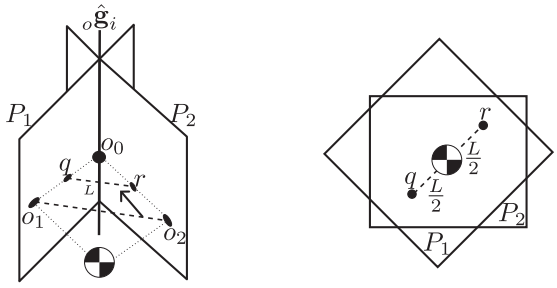


Fig. 6. Face-face pair with (a) intersecting faces and (b) coplanar faces.

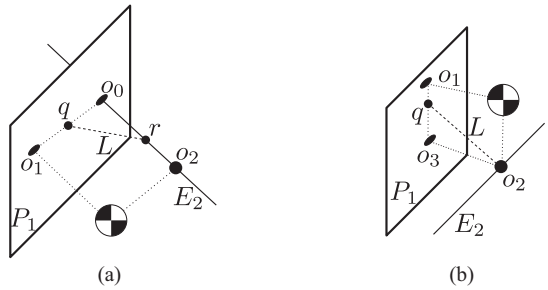


Fig. 7. Face-edge pair: (a) intersecting, (b) parallel.

Selecting Object Points: Several cases are considered depending on the relative orientation of the features. In each case, the points on the object are selected such that the distance between them is L and that the line joining them is as close as possible to the center of mass of the object.

A. Face-Face Pair: Two faces sampled on the object can be parallel, coplanar, or intersecting, as shown in Fig. 6.

Let two planes be defined by

$$P_1(d_1) : {}_o\hat{\mathbf{n}}_{f_1} \cdot \mathbf{p} = d_1 \tag{43}$$

$$P_2(d_2) : {}_o\hat{\mathbf{n}}_{f_2} \cdot \mathbf{p} = d_2 \tag{44}$$

where $\mathbf{p}$ is a point on the plane, ${}_o\hat{\mathbf{n}}_f$ is the normal of the face plane, and d is the distance of the plane from the origin. If the faces are parallel but not coplanar, the distance between the faces $(d_1 - d_2)_o\hat{\mathbf{n}}_{f_2}$ would need to be exactly equal to L for the match to be valid. This is almost never the case and would produce infinitesimal interpenetrations (i.e., friction) when placing the object.

A.1. Coplanar Faces: If the faces are coplanar, there is an infinite number of point pairs that are distanced by L. Among the possible pairs, we select one that minimizes the distance between the center of mass of the object and the line joining the two points to maximize the odds of sampling a stable pose. Let ${}_o\mathbf{u}$ be in the direction given by the extent of the object projected onto the common plane of the two faces. Selecting points along the direction vector that are at a distance $\pm L/2$ from the projection of the center of mass onto the face plane is done with

$${}_o\mathbf{p}_{o_1} = d_1 {}_o\hat{\mathbf{n}}_{f_1} + {}_o\mathbf{p}_c - ({}_o\mathbf{p}_c \cdot {}_o\hat{\mathbf{n}}_{f_1})\, {}_o\hat{\mathbf{n}}_{f_1} \tag{45}$$

$${}_o\mathbf{p}_q = {}_o\mathbf{p}_{o_1} + \frac{L}{2} {}_o\mathbf{u} \tag{46}$$

$${}_o\mathbf{p}_r = {}_o\mathbf{p}_{o_1} - \frac{L}{2} {}_o\mathbf{u} \tag{47}$$

where ${}_o\mathbf{p}_c$ is the center of mass of the object expressed in the object frame. In (45), ${}_o\mathbf{p}_{o_1}$ is the projection of the center of mass onto the face plane, while ${}_o\mathbf{p}_q$ and ${}_o\mathbf{p}_r$ are, respectively, the point on the first and second face being considered.

A.2. Intersecting Faces: If the faces are intersecting, they share a common edge whose direction is given by

$${}_o\hat{\mathbf{g}}_e = {}_o\hat{\mathbf{n}}_{f_1} \times {}_o\hat{\mathbf{n}}_{f_2} / \left\| {}_o\hat{\mathbf{n}}_{f_1} \times {}_o\hat{\mathbf{n}}_{f_2} \right\| \tag{48}$$

and whose origin ${}_o\mathbf{p}_e$ is defined by

$$\mathbf{u} = {}_o\hat{\mathbf{n}}_{f_1} \times {}_o\hat{\mathbf{g}}_e \tag{49}$$

$${}_o\mathbf{p}_e = {}_o\mathbf{p}_{o_1} + \frac{d_2 - {}_o\hat{\mathbf{n}}_{f_2} \cdot {}_o\mathbf{p}_{o_1}}{{}_o\hat{\mathbf{n}}_{f_2} \cdot \mathbf{u}} \mathbf{u} \tag{50}$$

where ${}_o\mathbf{p}_{o_1}$ is the projection of the center of mass onto the plane of the first face, as defined previously. The plane orthogonal to the common edge and passing through the center of mass of the object intersects ${}_o\hat{\mathbf{g}}_e$ at

$${}_o\mathbf{p}_{o_0} = {}_o\mathbf{p}_e + ({}_o\mathbf{p}_c \cdot {}_o\hat{\mathbf{g}}_e)\, {}_o\hat{\mathbf{g}}_e \tag{51}$$

such that a triangle is formed by ${}_o\mathbf{p}_{o_0}$, ${}_o\mathbf{p}_{o_1}$, and ${}_o\mathbf{p}_{o_2}$ in the plane passing through ${}_o\mathbf{p}_c$. A similar triangle connecting ${}_o\mathbf{p}_{o_0}$, ${}_o\mathbf{p}_q$, ${}_o\mathbf{p}_r$ whose leg joining both faces has length L can be defined with

$${}_o\mathbf{p}_q = {}_o\mathbf{p}_{o_0} + \frac{L}{\left\| {}_o\mathbf{p}_{o_1} - {}_o\mathbf{p}_{o_2} \right\|} ({}_o\mathbf{p}_{o_1} - {}_o\mathbf{p}_{o_0}) \tag{52}$$

$${}_o\mathbf{p}_r = {}_o\mathbf{p}_{o_0} + \frac{L}{\left\| {}_o\mathbf{p}_{o_1} - {}_o\mathbf{p}_{o_2} \right\|} ({}_o\mathbf{p}_{o_2} - {}_o\mathbf{p}_{o_0}) \tag{53}$$

where ${}_o\mathbf{p}_{o_2}$ is the projection of the center of mass onto the plane of the second face, and ${}_o\mathbf{p}_q$ and ${}_o\mathbf{p}_r$ are respectively the point on the first and second face being considered.

B. Face-Edge Pair: In the situation, where one interface is between two faces and the other interface is between a face and an edge, the line vector connecting the two contact points on the object will join a point on the face to a point on the edge, as shown in Fig. 7. Let the plane of the face/face interface be defined by

$$P_1(d) : {}_o\hat{\mathbf{n}}_f \cdot {}_o\mathbf{p}_q = d \tag{54}$$

and the edge be defined by

$$E_2(t) : {}_o\mathbf{p}_r = {}_o\mathbf{p}_e + t_o\mathbf{u}_e \tag{55}$$

such that $\mathbf{q}$ is constrained to lie on the plane of the face/face interface and $\mathbf{r}$ is constrained to lie on the edge.

The edge can either be parallel to the face plane or intersecting it. In both cases, a triangle is defined on a plane that is passing through the center of mass of the object to minimize the odds of sampling an unstable pose, as done in the face–face pair case.

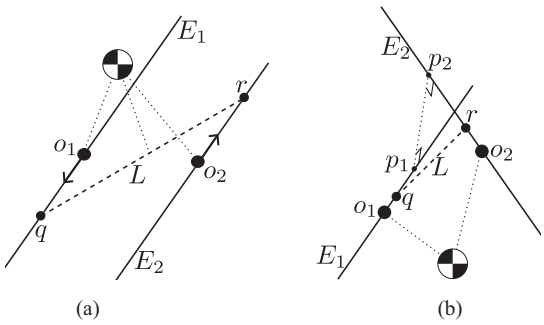


Fig. 8. Edge-edge pair: (a) parallel, (b) skew.

B.1. Intersecting: If the edge intersects the plane of the face, the intersection point ${}_o\mathbf{p}_{o_0}$ is given by

$$t = \frac{d - {}_o\hat{\mathbf{n}}_f \cdot {}_o\mathbf{p}_e}{{}_o\hat{\mathbf{n}}_f \cdot {}_o\mathbf{u}_e} \tag{56}$$

$${}_o\mathbf{p}_{o_0} = {}_o\mathbf{p}_e + t_o\mathbf{u}_e. \tag{57}$$

The projections of the center of mass onto the face and edge are given by

$${}_o\mathbf{p}_{o_1} = d_o\hat{\mathbf{n}}_f + {}_o\mathbf{p}_c - ({}_o\mathbf{p}_c \cdot {}_o\hat{\mathbf{n}}_f)\,{}_o\hat{\mathbf{n}}_f \tag{58}$$

$${}_o\mathbf{p}_{o_2} = {}_o\mathbf{p}_e + (({}_o\mathbf{p}_c - {}_o\mathbf{p}_e) \cdot {}_o\mathbf{u}_e)\,{}_o\mathbf{u}_e \tag{59}$$

respectively, such that a triangle is formed by ${}_o\mathbf{p}_{o_0}$, ${}_o\mathbf{p}_{o_1}$, and ${}_o\mathbf{p}_{o_2}$ in the plane passing through ${}_o\mathbf{p}_c$. The method employed in the face–face pair case can then be used to find ${}_o\mathbf{p}_q$ and ${}_o\mathbf{p}_r$.

B.2. Parallel: If the edge is parallel to the face plane, ${}_o\mathbf{p}_r$ is set to the projection of the center of mass onto the edge with

$${}_o\mathbf{p}_r = {}_o\mathbf{p}_{o_2}. \tag{60}$$

In the case of the edge not lying in the plane of the face, ${}_o\mathbf{p}_{o_2}$ is moved to the closest point on the face with

$${}_o\mathbf{p}_{o_3} = d_o\hat{\mathbf{n}}_f + {}_o\mathbf{p}_{o_2} - ({}_o\mathbf{p}_{o_2} \cdot {}_o\hat{\mathbf{n}}_f)\,{}_o\hat{\mathbf{n}}_f \tag{61}$$

where ${}_o\mathbf{p}_{o_1}$ and ${}_o\mathbf{p}_{o_2}$ are the projections of the center of mass onto the face and edge, respectively, as defined previously. The location of the point on the face is then given by

$${}_o\mathbf{p}_q = {}_o\mathbf{p}_{o_3} + \frac{\sqrt{\left(L^2 - \|{}_o\mathbf{p}_{o_3} - {}_o\mathbf{p}_{o_2}\|^2\right)}\,({}_o\mathbf{p}_{o_1} - {}_o\mathbf{p}_{o_3})}{\|{}_o\mathbf{p}_{o_1} - {}_o\mathbf{p}_{o_3}\|} \tag{62}$$

such that the projection of the center of mass onto the face plane lies on the line joining ${}_o\mathbf{p}_q$ to ${}_o\mathbf{p}_r$, as desired.

C. Edge-Edge Pair: The line vectors of two sampled edges can either be parallel, intersecting, or skew, as shown in Fig. 8. For two edge lines defined by their parametric equations

$$E_1(t_1) : {}_o\mathbf{p}_q = {}_o\mathbf{p}_{e_1} + t_{1o}\mathbf{u}_{e_1} \tag{63}$$

$$E_2(t_2) : {}_o\mathbf{p}_r = {}_o\mathbf{p}_{e_2} + t_{2o}\mathbf{u}_{e_2} \tag{64}$$

the lines are parallel if ${}_o\mathbf{u}_{e_1} \times {}_o\mathbf{u}_{e_2} = \mathbf{0}$.

Unless edges are parallel, there are two points, ${}_o\mathbf{p}_{p_1} = E_1(t_1)$ and ${}_o\mathbf{p}_{p_2} = E_2(t_2)$, that are a minimum distance apart. These points are given for parameters

$$t_1 = \frac{({}_o\mathbf{u}_{e_1} \cdot {}_o\mathbf{u}_{e_2})({}_o\mathbf{u}_{e_2} \cdot {}^o_{e_2}\mathbf{p}_{e_1}) - ({}_o\mathbf{u}_{e_1} \cdot {}^o_{e_2}\mathbf{p}_{e_1})}{({}_o\mathbf{u}_{e_1} \cdot {}_o\mathbf{u}_{e_2})(1 - {}_o\mathbf{u}_{e_1} \cdot {}_o\mathbf{u}_{e_2})} \tag{65}$$

$$t_2 = \frac{({}_o\mathbf{u}_{e_2} \cdot {}^o_{e_2}\mathbf{p}_{e_1}) - ({}_o\mathbf{u}_{e_1} \cdot {}^o_{e_2}\mathbf{p}_{e_1})({}_o\mathbf{u}_{e_1} \cdot {}_o\mathbf{u}_{e_2})}{({}_o\mathbf{u}_{e_1} \cdot {}_o\mathbf{u}_{e_2})(1 - {}_o\mathbf{u}_{e_1} \cdot {}_o\mathbf{u}_{e_2})} \tag{66}$$

respectively, where ${}^o_{e_2}\mathbf{p}_{e_1} = {}_o\mathbf{p}_{e_1} - {}_o\mathbf{p}_{e_2}$ is the vector joining the origins of the two edges [39]. If the distance between the two closest points is nonzero, the two edges are skew. The projection of the center of mass onto E_1 and E_2 is given by

$${}_o\mathbf{p}_{o_1} = {}_o\mathbf{p}_{e_1} + (({}_o\mathbf{p}_c - {}_o\mathbf{p}_{e_1}) \cdot {}_o\mathbf{u}_{e_1})\,{}_o\mathbf{u}_{e_1} \tag{67}$$

$${}_o\mathbf{p}_{o_2} = {}_o\mathbf{p}_{e_2} + (({}_o\mathbf{p}_c - {}_o\mathbf{p}_{e_2}) \cdot {}_o\mathbf{u}_{e_2})\,{}_o\mathbf{u}_{e_2} \tag{68}$$

respectively.

C.1. Parallel Edge Lines: Similar to previous scenarios, the objective is to select points such that the distance between the object center of mass and the line joining the two points is minimized, to increase the odds of sampling a stable pose.

For parallel lines, the points ${}_o\mathbf{p}_q$ and ${}_o\mathbf{p}_r$ are selected by being equally distanced from their respective center of mass projections in opposite directions with

$${}_o\mathbf{p}_q = {}_o\mathbf{p}_{o_1} + \frac{1}{2}\sqrt{L^2 - \|{}_o\mathbf{p}_{o_1} - {}_o\mathbf{p}_{o_2}\|^2}\,{}_o\mathbf{u}_{e_1} \tag{69}$$

$${}_o\mathbf{p}_r = {}_o\mathbf{p}_{o_2} - \frac{1}{2}\sqrt{L^2 - \|{}_o\mathbf{p}_{o_1} - {}_o\mathbf{p}_{o_2}\|^2}\,{}_o\mathbf{u}_{e_1} \tag{70}$$

such that the two points are distanced by L. If $L < \|{}_o\mathbf{p}_{o_1} - {}_o\mathbf{p}_{o_2}\|$, ${}_o\mathbf{p}_q$ and ${}_o\mathbf{p}_r$ are set to the projections of the center of mass onto the edges.

C.2. Intersecting Edge Lines: If the edges are intersecting, the intersection point will form a triangle with points ${}_o\mathbf{p}_q$ and ${}_o\mathbf{p}_r$ on the edges. However, the two edges are not necessarily part of the same face on nonconvex objects.

The intersection point of the two edges ${}_o\mathbf{p}_{o_0}$ can be obtained with (65) by setting either t_1 or t_2 in its respective line equation. The points ${}_o\mathbf{p}_q$ and ${}_o\mathbf{p}_r$ can be obtained with (52) from ${}_o\mathbf{p}_{o_0}$, and the points ${}_o\mathbf{p}_{o_1}$, ${}_o\mathbf{p}_{o_2}$ in (67) and (68).

C.3. Skew Edge Lines: With skew edges, the closest point on each edge to the other edge, given by (65), define a line vector ${}^o_{p_2}\mathbf{p}_{p_1} = {}_o\mathbf{p}_{p_1} - {}_o\mathbf{p}_{p_2}$ that is orthogonal to both edges. Hence, edges lie in parallel planes that are orthogonal to ${}^o_{p_2}\mathbf{p}_{p_1}$ and separated by a distance $\|{}^o_{p_2}\mathbf{p}_{p_1}\|$. Therefore, the distance between $E_1(t_1)$ and $E_2(t_2)$ is given by

$$\|E_2(t_2) - E_1(t_1)\| = \sqrt{D^2 + \left\|{}^o_{p_2}\mathbf{p}_{p_1}\right\|^2} \tag{71}$$

where D is the distance along the parallel planes, such that the distance between ${}_o\mathbf{p}_{o_1}$ and ${}_o\mathbf{p}_{o_2}$ along the parallel planes is given by $\sqrt{\|{}^o_{o_2}\mathbf{p}_{o_1}\|^2 - \|{}^o_{p_2}\mathbf{p}_{p_1}\|^2}$. Similar to how the position of the contact points were defined in (52), the points ${}_o\mathbf{p}_q$ and ${}_o\mathbf{p}_r$ are defined with

$${}_o\mathbf{p}_q = {}_o\mathbf{p}_{p_1} + \frac{({}_o\mathbf{p}_{o_1} - {}_o\mathbf{p}_{p_1})\sqrt{L^2 - \left\|{}^o_{p_2}\mathbf{p}_{p_1}\right\|^2}}{\sqrt{\left\|{}^o_{o_2}\mathbf{p}_{o_1}\right\|^2 - \left\|{}^o_{p_2}\mathbf{p}_{p_1}\right\|^2}} \tag{72}$$

$$ {}_o\mathbf{p}_r = {}_o\mathbf{p}_{p_2} + \frac{({}_o\mathbf{p}_{o_2} - {}_o\mathbf{p}_{p_2})\sqrt{L^2 - \left\|{}^o_{p_2}\mathbf{p}_{p_1}\right\|^2}}{\sqrt{\left\|{}^o_{o_2}\mathbf{p}_{o_1}\right\|^2 - \left\|{}^o_{p_2}\mathbf{p}_{p_1}\right\|^2}} \tag{73} $$

that will produce a solution unless $\|{}^o_{p_2}\mathbf{p}_{p_1}\| > L$, in which case the edges are too far apart to produce a valid match.,

C. Determination of Object Pose

Let ${}_o\hat{\mathbf{n}}_{e_1}$ be equal to the average of the two face normals that E_1 is joining. By definition, ${}_o\hat{\mathbf{n}}_{e_1}$ will be in the plane orthogonal to E_1. For a placement to be valid, the normal at the scene contact point ${}_w\hat{\mathbf{n}}_a$ must also lie in the plane orthogonal to the edge. Moreover, the angle between ${}_o\hat{\mathbf{n}}_{e_1}$ and ${}_w\hat{\mathbf{n}}_a$ must be less than 90° for the placement to be nonpenetrating. Unless the two edges are collinear, we have that ${}^w_b\mathbf{p}_a \times {}_w\hat{\mathbf{n}}_a \neq 0$ such that there exists a vector

$$ {}_w\hat{\mathbf{u}} = \frac{{}^w_b\mathbf{p}_a \times {}_w\hat{\mathbf{n}}_a}{\|{}^w_b\mathbf{p}_a \times {}_w\hat{\mathbf{n}}_a\|} \tag{74} $$

that is orthogonal to the scene face and to ${}^w_b\mathbf{p}_a$, the direction of the line vector connecting the two contact points. The vector in the object frame that is analogous to ${}_w\hat{\mathbf{u}}$ is given by

$$ {}_o\hat{\mathbf{u}} = -\frac{{}^o_r\mathbf{p}_q \times {}_o\hat{\mathbf{n}}_{e_1}}{\|{}^o_r\mathbf{p}_q \times {}_o\hat{\mathbf{n}}_{e_1}\|} \tag{75} $$

such that

$$ {}_w\hat{\mathbf{u}} = {}_w\mathbf{R}_o\,{}_o\hat{\mathbf{u}} \tag{76} $$

relates the two vectors.

Defining basis vectors

$$ {}_w\hat{\mathbf{w}} = \frac{{}_w\hat{\mathbf{u}} \times {}^w_b\mathbf{p}_a}{\|{}_w\hat{\mathbf{u}} \times {}^w_b\mathbf{p}_a\|} \tag{77} $$

$$ {}_o\hat{\mathbf{w}} = \frac{{}_o\hat{\mathbf{u}} \times {}^o_r\mathbf{p}_q}{\|{}_o\hat{\mathbf{u}} \times {}^o_r\mathbf{p}_q\|} \tag{78} $$

a frame fixed in the scene can be defined with

$$ \boldsymbol{\mathcal{F}}_s = \begin{bmatrix} {}_w\hat{\mathbf{u}} & \frac{{}^w_b\mathbf{p}_a}{\|{}^w_b\mathbf{p}_a\|} & {}_w\hat{\mathbf{w}} \end{bmatrix}^{\mathsf{T}}. \tag{79} $$

A frame fixed in the object can be obtained with

$$ \boldsymbol{\mathcal{F}}_o = \begin{bmatrix} {}_o\hat{\mathbf{u}} & \frac{{}^o_r\mathbf{p}_q}{\|{}^o_r\mathbf{p}_q\|} & {}_o\hat{\mathbf{w}} \end{bmatrix}^{\mathsf{T}} \tag{80} $$

such that

$$ {}_w\mathbf{R}_o = \boldsymbol{\mathcal{F}}_s\boldsymbol{\mathcal{F}}_o^{\mathsf{T}} \tag{81} $$

is the orientation of the object in the world frame. With the knowledge of ${}_w\mathbf{R}_o$, the position of the object can subsequently be obtained with

$$ {}_w\mathbf{p}_o = {}_w\mathbf{p}_a - {}_w\mathbf{R}_o\,{}_o\mathbf{p}_q \tag{82} $$

where ${}_w\mathbf{p}_a$ is the position of the first scene contact point.

D. Pose Validation

Once a pose is determined, it undergoes a series of checks to ensure that the object does not interpenetrate other objects and does not destabilize the assembly. First, a collision detection algorithm verifies that the object avoids penetrating other assembly components, and detects contact interfaces on which the object is resting. To determine the location of contact points, we use the approach from [8] and select the vertices of the convex hull over the contact interface as contact points, as shown in Fig. 3. Reaction forces at the contact points are then computed using the QR-based method in (8) and the maximal tension force is compared to a threshold value (e.g., 5 Newtons) to avoid performing the more computationally expensive QP-based method when the placement is clearly unstable. Finally, the forces at the contact points are computed using (5), the equilibrium of the assembly is verified, and the static robustness of the assembly is updated.

VI. Simulation Experiments

We perform a series of simulation experiments across six different scenes, shown in Fig. 9, where the task is to place a cube such that it stably rests on other objects. Each scene is designed to challenge the algorithms in different ways. The *Stack* scene has many points on vertical surfaces, *Pyramids* has few robust contact points, *Table* is prone to toppling, *Sawteeth* has no horizontal surfaces, and *Canyon* only affords placements resting on both objects. The benchmarked algorithms and evaluation criteria are described in Sections VI-A and Section VI-B, respectively, and the results are reported in Table I. For all sample-based planners, 50 consecutive experiments are performed for each scene with a maximum of 500 attempts per experiment. Algorithms are evaluated in terms of planning time, assembly robustness, packing density, and success rate.

A. Benchmarked Algorithms

1) Planner Using Static Robustness (Ours-SR): We compare to other methods our proposed algorithm defined in Section V and using (29) to sample points, with the initial probability of sampling the most robust point set to 10% and the exponential decay rate λ set to 0.99. In all experiments involving contact point sampling, if multiple pairs of features can be matched to the sample points, a single random pair is selected.

2) Planner With Uniform Sampling (Ours-Uniform): In essence, our algorithm, defined in Section V, uses the proposed static robustness heuristic to increase the odds of sampling promising contact points in the scene. Insights about the effectiveness of the proposed algorithm can be obtained by comparing it with a variation for which the probability of sampling a point is uniform. If there is a significant benefit to using the static robustness heuristic, the proposed algorithm should perform better when the probability distribution in (29) is used, and worst when it is uniform.

3) Dynamics Simulation From Above (Sim-Above): A dynamics simulator is also used to provide a standard for comparison. Object placement from simulations is performed in two phases during which the restitution coefficient is set to zero, and the dynamics are heavily damped to reduce any bouncing phenomena. In the first phase, the object is dropped in a random orientation with its center of mass at a random position slightly

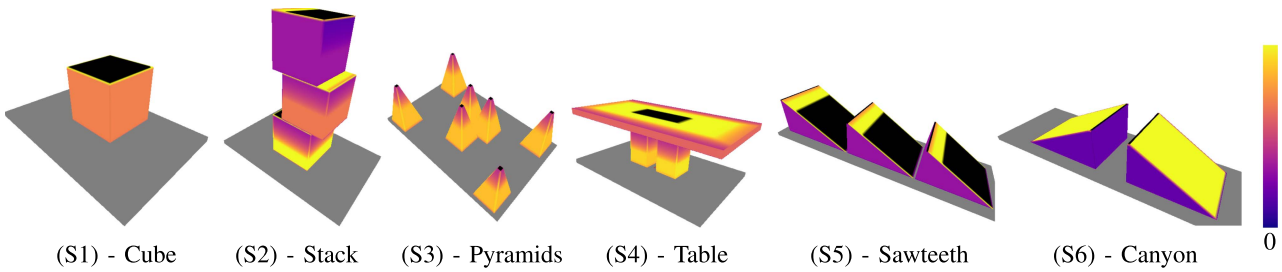


Fig. 9. Six scenes used for the experiments. The surface color indicates the relative magnitude of the maximal normal force that can be sustained, with a lighter color indicating a larger force and black indicating an infinite force.

TABLE I
AVERAGE PLACEMENT TIME IN SECONDS, ROBUSTNESS TO EXTERNAL WRENCH (ROB.) IN NEWTONS, MINIMUM STATIC ROBUSTNESS (MIN. SR) IN NEWTONS, AND VOLUME OF THE ASSEMBLY IN CUBIC METERS FOR EACH ALGORITHM ACROSS OUR SIX SCENES

		Chance	Sim-Random	Sim-Above	HM	Ours-Uniform	Ours-SR
Cube	Time	2.85	11.23	0.79	0.88	3.60	**0.38**
	Rob.	—	1.72	1.99	**3.21**	1.64	1.81
	Min. SR	—	2.29	2.64	**4.44**	2.21	2.47
	Volume	—	30.0	28.4	**17.2**	30.1	28.0
Stack	Time	4.88	54.5	7.76	**0.56**	120	1.51
	Rob.	—	0.48	0.41	**0.80**	0.53	0.56
	Min. SR	—	1.76	1.88	**2.40**	2.10	2.00
	Volume	—	70.6	66.7	**47.3**	58.0	61.1
Table	Time	5.13	20.8	6.59	0.83	13.8	**0.40**
	Rob.	—	1.76	1.85	1.11	1.86	**1.95**
	Min. SR	—	2.10	2.10	2.10	1.65	2.10
	Volume	—	105	103	109	**88.3**	101
Pyramids	Time	6.47	97.25	41.74	**1.10**	18.62	1.71
	Rob.	—	0.78	0.75	**0.85**	0.49	0.40
	Min. SR	—	**1.69**	1.10	1.54	0.41	0.42
	Volume	—	80.1	**72.4**	128	96.8	135
Sawteeth	Time	4.30	9.43	8.74	30.0	10.4	**2.86**
	Rob.	—	0.71	0.64	—	0.91	0.77
	Min. SR	—	1.91	1.80	—	1.58	1.73
	Volume	—	**136**	139	—	161	159
Canyon	Time	4.02	55.2	45.7	31.6	9.85	**1.30**
	Rob.	—	0.85	0.88	—	1.40	**1.68**
	Min. SR	—	0.57	0.29	—	0.52	**0.72**
	Volume	—	109	**108**	—	130	128
Average Time		4.61	41.40	18.55	10.83	29.38	**1.36**
Success Rate		0%	93.3%	99.7%	66.7%	95.3%	**100%**

Entries with — indicate that no valid placement was found in the experiment.

above the nonfixed objects in the scene. The gravitational acceleration is set to $0.1\ \mathrm{m/s^2}$ and the downward velocity of the object is set to $0.01\ \mathrm{m/s}$ to reduce the kinetic energy of the object when it collides with the scene. The simulation is run until the object comes to rest and is aborted if the object falls off the scene floor. In the second phase, gravity is increased to $9.81\ \mathrm{m/s^2}$ and the simulation is ran for 100 iterations to ensure that the placement is stable under standard gravitational acceleration. If the placed object rests on at least one other object, and no object has moved significantly, the placement is considered to be successful.

4) Dynamics Simulation From a Random Pose (Sim-Random): Performing simulations from initial poses above the scene objects will not allow the planner to find placement poses below other objects. A more general approach is to perform simulations from random poses, where the pose of the object is initialized randomly within the vicinity of the scene objects. When sampling positions, the extent of the scene was defined from the bounding box of the objects in the scene and points around the scene, up to a distance equal to the radius of the sphere circumscribing the object, were considered. Otherwise, all parameters are kept the same as in the *Sim-above* algorithm.

5) Heightmap Minimization (HM): The algorithm defined in [40] is implemented and used as a benchmark. A summary of the algorithm is provided below. Planning with the *HM* algorithm is performed in four steps: 1) candidate pose generation, 2) scoring, 3) stability evaluation, and potentially 4) refined search. The first step uses [41] to generate four object orientations that are likely stable on planar surfaces, and perturb the orientations with rotations about the vertical axis. The scene is voxelized and a set of candiate poses is generated by finding the lowest

point in the voxel grid for each object orientation. The second step scores the candidate poses with a heightmap minimization heuristic proposed in [40] that encourages dense and stable placements. The third step iterates over the highest scoring poses and evaluates their stability by solving (5) and returning the first stable pose found. Finally, in the event that no stable pose is found, the search is refined by generating a larger set of candidate poses through rotation perturbations about orthogonal axes in the horizontal plane. Our implementation of *HM* uses the same parameters and the same voxel resolution as in [40].

6) Random Pose (Chance): To provide a baseline for comparison, a random pose within the vicinity of the scene objects is selected for the object, and the stability of the placement is evaluated. Although this method is expected to be the slowest, it should be capable of finding stable placements if enough iterations are performed (although a maximum of 500 attempts are made in the experiments).

B. Evaluation Criteria

Five criteria are used to evaluate the performance of the algorithms.

1) Placement Time: For each set of experiments, the average time required to find a collision-free stable placement pose is measured. Since some algorithms may require more iterations but less compute time per iteration, all algorithms are compared in terms of the average compute time per valid placement.

2) Assembly Robustness: The objective of inertia-aware object placement planning is to find placements that maximize the force required to displace any object in the assembly. Hence, the minimal magnitude of the external wrench that would destabilize the assembly is computed according to [5], [42], where the wrench directions are defined as the orthogonal axes in the 6D wrench space.

3) Minimum Static Robustness: Our static robustness heuristic defined in Section IV is computed for points on all objects in the scene following a valid placement. The minimum static robustness, representing the smallest force that can be applied to the surface of an object such that the assembly is destabilized, is computed for each experiment.

4) Packing Density: Since it is often desired to place objects such that the assembly is dense, the packing density of the assembly is computed as the volume of the oriented bounding box of the assembly following the placement.

5) Success Rate: The proportion of experiments in which a valid placement was found is reported for each algorithm.

VII. Real Robot Experiments

A practical placement planner should produce stable assemblies under the presence of uncertainty in the pose, shape, and inertial parameters of the objects. To evaluate the effectiveness of the proposed algorithm in such a scenario, we performed experiments with a Franka Research 3 robot arm equipped with a Robotiq 2F-85 gripper. A set of 10 nonconvex basswood objects, pictured in Fig. 10, is used for this experiment, where each object is built from manually glued cuboids resulting in shapes that

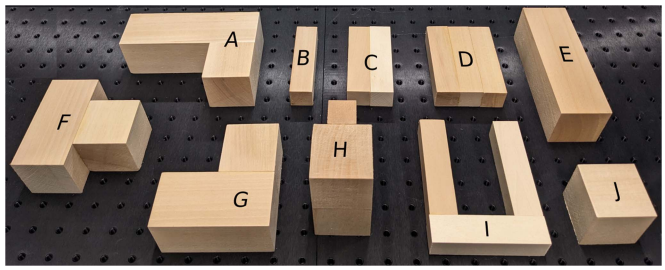


Fig. 10. Set of 10 basswood objects used for the real robot experiments.

are slightly different from the models used by the planner. The mass density of the objects is assumed to be 415 $\mathrm{kg/m^3}$, which is expected to differ from the actual value, introducing errors in the inertial models of the objects. In our experiments, the friction coefficient was assumed to be 0.5 at interfaces between our basswood objects, which is expected to be an optimistic value. In practical applications where stability is critical, the friction coefficient can be assumed to be lower than its nominal value such that the planner acts more conservatively. Uncertainty in the initial pose of the objects, slippage, and Cartesian trajectory errors are expected to create a discrepancy between the planned and executed placements, potentially resulting in unstable assemblies.

A total of 50 placements from 10 experiments pictured in Fig. 11 (E1 to E10) are performed with the robot. Each experiment consists of iteratively placing objects in the scene until five objects are placed. For each placement, the proposed planner is ran 10 times, each time executing at most 20 iterations with an object selected randomly without replacement and with odds proportional to its mass (i.e., heavier objects are more likely to be placed first). For these experiments, sampling on the fixed support is allowed and (33) is used with $\gamma = 0.5$ and with other parameters set to the same values as in the simulation experiments. If more than one placement results in the same minimum SR, the one with the largest median SR is selected in E1–E5, and the one with the smallest assembly volume is selected in E6–E10.

With the placement pose defined, a feasible grasp as far as possible to the contact points is determined. For each placement, the RRTConnect [43] motion planner was run twice, each time for at most 10 s, to find collision-free trajectories to pick up and place the object.

The success of any placement depends on five cascading steps: 1) finding a feasible placement pose, 2) grasping the object, 3) planning a collision-free motion trajectory, 4) executing the placement without slippage or collisions, and 5) keeping the assembly stable. During our experiments, human observation was used to determine the outcome of each step and any issue that arose (e.g., slip, contact with the environment, toppled assembly) was noted. The human observer manually carried out any uncompleted placement by approximately placing the object in its goal pose such that the following placements could be subsequently performed by the robot. The success rate of each

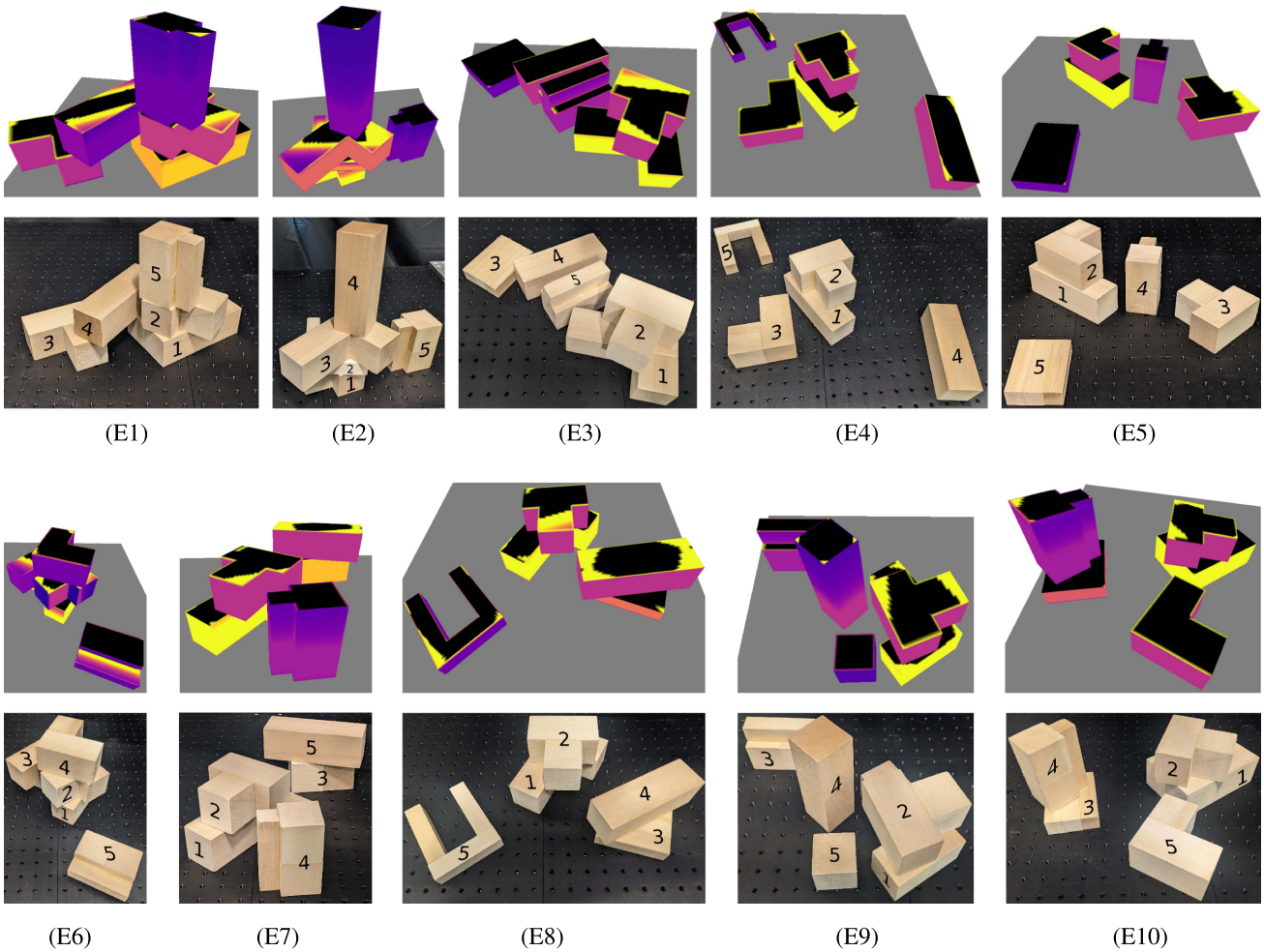


Fig. 11. Ten scenes from the real robot experiments with the SR map in the top row and the final assembly in the bottom row. The order in which objects are placed is indicated by the numbers on the objects.

TABLE II
SUCCESS RATE OF EACH STEP IN THE REAL ROBOT EXPERIMENTS WITH 50 PLACEMENTS, COMPUTED RELATIVE TO THE NUMBER OF SUCCESSFUL ATTEMPTS FROM THE PREVIOUS STEP

Step	Success Rate
Placement Planning	50/50 = 100%
Grasping	41/50 = 82%
Motion Planning	37/41 = 90.2%
Placement Execution	33/37 = 89.2%
Stable Assembly	33/33 = 100%

step relative to the number of placements that reached the step is reported in Table II.

VIII. DISCUSSION

The performance of the proposed algorithm in comparison with other algorithms is analyzed in Section VIII-A. Observations made during the real robot experiments are outlined in Section VIII-B and the computational complexity of the proposed algorithm is analyzed in Section VIII-C.

A. Performance Analysis

In our experiments, *Chance* never found a valid placement in the 3000 attempts made, indicating that the placement planning problem is not trivial.

According to Table I, the proposed algorithm using static robustness was about 20 times faster to find a stable placement when compared to the same algorithm without static robustness and about eight times faster than the *HM* algorithm. The simulator also took much longer than the proposed algorithm to find a valid solution, on average taking about 13.6 times longer with *Sim-Above* and 30 times longer with *Sim-Random*. This poor performance is expected, given that the simulator needlessly solves a much more complex problem (i.e., coupled kinematics and dynamics) [28].

Compared to other algorithms, *HM* can find placements that are more robust to external wrenches. However, *HM* found valid placements in only 66.7% of the experiments, while the other algorithms (except *Chance*) found valid placements in 93% to 100% of the experiments. Even though *HM* optimizes for packing density while *Ours-SR* does not, both algorithms produce assemblies with similar volumes. In contrast, the other

algorithms can produce better packed assemblies by having the object be supported by the floor (e.g., under the table).

Algorithms that sample uniformly struggle in large scenes that afford only few placements, as seen in the *Stack* scene where the time required by *Sim-Random* and *Ours-Uniform* is significantly higher than for the other algorithms. In the *Table* scene, the robustness to external wrench is significantly lower for *HM* since it selects the first stable pose found, which is not the one in the center of the table. However, in *Pyramids*, *Ours-SR* finds placements with relatively low robustness compared to *HM* since the former does not center the cube on the three pyramids. By virtue of the three pyramids being identical, *HM* produces better centered placements, resulting in a more robust assembly. In the *Sawteeth* and *Canyon* scenes, *HM* cannot find any valid placements since it iterates over the N highest scoring placements, which are all unstable in these scenes. This issue could be alleviated by performing stability checks at the candidate pose generation stage, at the cost of a prohibitive increase in computation time due to *HM*'s iteration over all discretized positions and orientations of the object.

B. Observations in Real Robot Experiments

Four objects in the set (B, C, D, and I) are relatively long but thin, making them quite weak when placed upright and quite robust when placed lying down. Two of these objects (D and I) were frequently selected by the planner in the experiments (i.e., in E2–E5, E8, E10), and a lying down pose was always selected for them, which is expected from a SR point of view. However, the objects being thin, the gripper could not grasp them when they were lying down, ultimately resulting in a grasp planning failure. Similarly, although the pose selected for object H in E3 and object J in E9 are collision-free and make sense from a SR point of view, placing them would have caused the gripper to collide with other objects in the scene. This issue suggests that the planning of the grasping and release phases should be better integrated with placement planning. A rejection sampling step as done in [5], [40] could be performed to ensure that the object can be grasped and placed in the desired pose. Also, scene and object points that generated a grasp planning failure could have their likelihood of being selected again reduced in subsequent iterations with a slight modification to (29). In two occasions, the motion planner could not find a path to the placement pose in the allocated time, highlighting that planning for placements that can be easily executed is an avenue for future work.

A supplemental video, from which Fig. 1 is taken, is provided[2] to illustrate the real robot experiments. The accompanying video highlights 1) a case where the robot successfully places objects in a scene and 2) another where the robot topples the assembly when placing the last object. In the first case, when the robot placed the last object whose static robustness is deemed to be very low, a slight pose error almost caused the neighbouring object to topple, effectively validating the computed static robustness. In the second case, the robot created a contact with a neighboring object when placing the last object due to an erroneous motion trajectory. This contact exerted force in a direction in which an infinite amount of force could be sustained according to the computed static robustness. As the robot applied force, the whole structure collapsed, outlining the importance of using static robustness not only to define the placement pose, but also the approach direction of the robot when placing an object.

[2]At the following URL: http://tiny.cc/stableplacement

TABLE III
AVERAGE COMPUTE TIME PER ITERATION FOR EACH STEP

	Compute Time (ms)	Proportion
QR-based Reaction Forces	9.8	6.8%
QP-based Reaction Forces	82.3	57.3%
Static Robustness	15.0	10.4%
Pose Definition	0.6	0.4%
Contact Resolution	35.9	25.0%
Total	144	100%

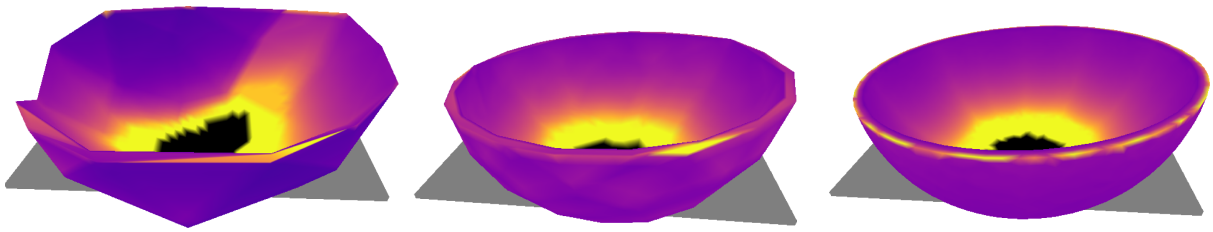
C. Computational Complexity

As expected, in terms of compute time, the contact resolution and constrained optimization steps are the most demanding, taking together more than 80% of the total compute time according to Table III. The QR-based reaction force solver being significantly faster than the QP-based solver confirms that the latter should be used only to verify placements that are likely to be successful. Per iteration, the overhead of computing the static robustness heuristic accounts for about 10% of the compute time, which is largely outweighed by the time saved with the reduction in the number of iterations required to find a solution.

By design, the computation time of the SR map grows linearly with the number of objects in the scene. This is due to each object being considered in isolation in the computation of the SR map. Hence, in a scene comprising a large number of objects, the proposed heuristic should enable our placement planner to focus on subsets of the scene that are more likely to offer a strong support.

Objects with curved shapes can be challenging for the proposed algorithm, which relies on planar interfaces. The shape of a curved object, like the bowl shown in Fig. 12, can be approximated by a triangle mesh, with a lower number of vertices producing a coarser approximation that might result in erroneous placement planning. While a higher number of vertices usually reduces the modeling error, it also increases the computational cost of the algorithm and ultimately slows down the planning process.

Placement planning experiments on a hemispherical shell with flattened end (i.e., a bowl), were performed to study the computational complexity of the proposed algorithm. Triangle meshes with various numbers of vertices were obtained through quadric decimation [44] and used by the proposed algorithm to find stable placements for a cube. Ten experiments were performed for each mesh, and the average time required to compute the SR map and find a stable placement was recorded. Results from these experiments are shown in Fig. 13, where the time required to compute the SR map grows approximately



(a) (b) (c)

Fig. 12. Bowl, with various vertex number and overlaid SR map, used to study the computational complexity of the proposed algorithm with curved objects approximated by triangle meshes. (a) 50 vertices. (b) 150 vertices. (c) 1500 vertices.

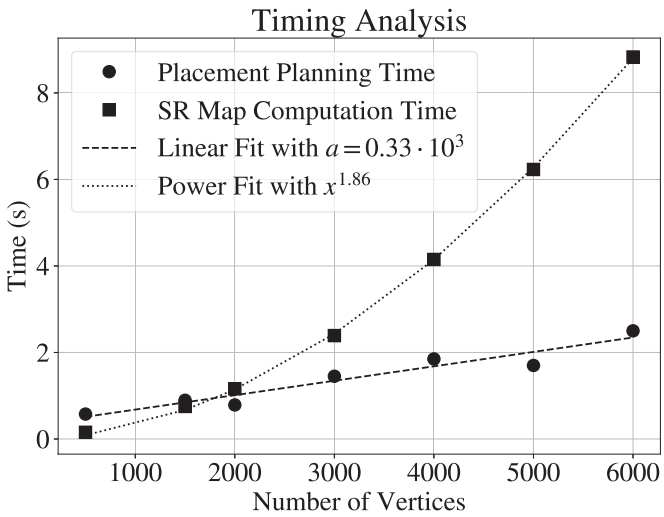


Fig. 13. Time required to compute the SR map grows less than quadratically with the number of vertices in the object mesh, and the time required to find a stable placement grows linearly with the number of vertices.

following:

$$t_{\text{SR}} = 8.53 \cdot 10^{-7} \cdot v^{1.856} \tag{83}$$

and the time required to find a stable placement grows approximately following:

$$t_{\text{plan}} = 0.346 + 0.333 \cdot 10^{3} v \tag{84}$$

where v is the number of vertices in the object mesh. Hence, the time required to compute the SR map grows less than quadratically with the number of vertices in the object mesh, and the time required to find a stable placement grows linearly with the number of vertices. However, as can be noticed in Fig. 12, there is a diminishing return to increasing the resolution of the SR map beyond a certain point with only a subtle difference between Fig. 12(b) and (c), with the latter computed at a $10\times$ greater resolution. Consequently, an adequate mesh resolution should be selected to balance the trade-off between modeling error and computational complexity. Although the execution time of the proposed algorithm does not grow prohibitively with the number of vertices, the processing time can be limited by using a coarser resolution for SR computations and an interpolation scheme to produce a finer map.

IX. Conclusion

This work defines an inertia-aware object placement planner that can scale to scenes with many objects with no assumption made on object convexity, homogeneous density, or shape. Our algorithm makes use of the object inertial parameters at every step of the process to increase the odds of sampling a stable placement pose and to preserve the assembly's robustness to external perturbations. We show that the use of our *static robustness* heuristic greatly increases the odds of sampling a stable placement pose, with a light computational overhead, making the proposed algorithm much faster than other methods. Future work will focus on improving the efficiency of the algorithm by performing a local optimization after having defined a placement pose such that an almost stable pose can be adjusted to be stable. Also, a learning-based approach to static robustness inference will be investigated to improve compute time while keeping the generalizability of the method.

References

[1] R. Alterovitz, S. Koenig, and M. Likhachev, "Robot planning in the real world: Research challenges and opportunities," *AI Mag.*, vol. 37, no. 2, pp. 76–84, 2016.

[2] K. Srinivas et al., "The busboy problem: Efficient tableware decluttering using consolidation and multi-object grasps," in *Proc. IEEE Int. Conf. Automat. Sci. Eng.*, 2023, pp. 1–6.

[3] M. Wermelinger, R. Johns, F. Gramazio, M. Kohler, and M. Hutter, "Grasping and object reorientation for autonomous construction of stone structures," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5105–5112, Jul. 2021.

[4] D. Baraff, "Coping with friction for non-penetrating rigid body simulation," *ACM SIGGRAPH Comput. Graph.*, vol. 25, no. 4, pp. 31–41, 1991.

[5] H. Chen, W. Wan, K. Koyama, and K. Harada, "Planning to build block structures with unstable intermediate states using two manipulators," *IEEE Trans. Automat. Sci. Eng.*, vol. 19, no. 4, pp. 3777–3793, Oct. 2021.

[6] J. A. Haustein, K. Hang, J. Stork, and D. Kragic, "Object placement planning and optimization for robot manipulators," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 7417–7424.

[7] R. Mattikalli, P. K. Khosla, B. Repetto, and D. Baraff, "Stability of assemblies," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1993, vol. 1, pp. 652–661.

[8] R. Mattikalli, D. Baraff, and P. Khosla, "Finding all stable orientations of assemblies with friction," *IEEE Trans. Robot. Automat.*, vol. 12, no. 2, pp. 290–301, Apr. 1996.

[9] J.-S. Pang and J. C. Trinkle, "Stability characterizations of fixtured rigid bodies with coulomb friction," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, vol. 1, pp. 361–368.

[10] E. Whiting, H. Shin, R. Wang, J. Ochsendorf, and F. Durand, "Structural optimization of 3D masonry buildings," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 1–11, 2012.

[11] G. T.-C. Kao et al., "Coupled rigid-block analysis: Stability-aware design of complex discrete-element assemblies," *Comput.-Aided Des.*, vol. 146, 2022, Art. no. 103216.
[12] H. Goldstein, C. Poole, and J. Safko, *Classical Mechanics*, 3rd ed., Singapore: World Scientific, 2002.
[13] H. V. Shin, C. F. Porst, E. Vouga, J. Ochsendorf, and F. Durand, "Reconciling elastic and equilibrium methods for static analysis," *ACM Trans. Graph.*, vol. 35, no. 2, pp. 1–16, 2016.
[14] C. Borst, M. Fischer, and G. Hirzinger, "A fast and robust grasp planner for arbitrary 3D objects," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1999, vol. 3, pp. 1890–1896.
[15] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1992, vol. 3, pp. 2290–2295.
[16] W. C. Agboh et al., "Learning to efficiently plan robust frictional multi-object grasps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 10660–10667.
[17] L. Kavraki, J.-C. Latombe, and R. H. Wilson, "On the complexity of assembly partitioning," *Inf. Process. Lett.*, vol. 48, no. 5, pp. 229–235, 1993.
[18] M. Toussaint, "Logic-geometric programming: An optimization-based approach to combined task and motion planning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 1930–1936.
[19] D. Xu, R. Martín-Martín, D.-A. Huang, Y. Zhu, S. Savarese, and L. F. Fei-Fei, "Regression planning networks," *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 1319–1329, 2019.
[20] Y. Zhu, J. Tremblay, S. Birchfield, and Y. Zhu, "Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 6541–6548.
[21] Y. Jiang, M. Lim, C. Zheng, and A. Saxena, "Learning to place new objects in a scene," *Int. J. Robot. Res.*, vol. 31, no. 9, pp. 1021–1043, 2012.
[22] Y. Lee, W. Thomason, Z. Kingston, and L. E. Kavraki, "Object reconfiguration with simulation-derived feasible actions," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 8104–8111.
[23] T. Motoda, D. Petit, T. Nishi, K. Nagata, W. Wan, and K. Harada, "Shelf replenishment based on object arrangement detection and collapse prediction for bimanual manipulation," Robotics, vol. 11, no. 5, 2022, Art. no. 104.
[24] R. Kartmann, F. Paus, M. Grotz, and T. Asfour, "Extraction of physically plausible support relations to predict and validate manipulation action effects," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3991–3998, Oct. 2018.
[25] P. Nadeau, M. Giamou, and J. Kelly, "The sum of its parts: Visual part segmentation for inertial parameter identification of manipulated objects," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 3779–3785.
[26] B. Sundaralingam and T. Hermans, "In-hand object-dynamics inference using tactile fingertips," *IEEE Trans. Robot.*, vol. 37, no. 4, pp. 1115–1126, Aug. 2021.
[27] P. Nadeau, "A standard rigid transformation notation convention for robotics research," 2024, *arXiv:2405.07351*.
[28] D. M. Kaufman, "Coupled principles for computational frictional contact mechanics," PhD Thesis, Rutgers University Graduate School, New Brunswick, NJ, USA, 2009.
[29] I. Kao, K. M. Lynch, and J. W. Burdick, "Contact modeling and manipulation," in *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2016, pp. 931–954.
[30] D. Baraff, "Analytical methods for dynamic simulation of non-penetrating rigid bodies," in *Proc. Conf. Comput. Graph. Interactive Techn.*, 1989, pp. 223–232.
[31] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, pp. 25–57, 2006.
[32] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: JHU Press, 2013.
[33] P. R. Sinha and J. M. Abel, "A contact stress model for multifingered grasps of rough objects," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1990, pp. 1040–1045.
[34] B. Bhushan, "Contact mechanics of rough surfaces in tribology: Multiple asperity contact," *Tribol. Lett.*, vol. 4, pp. 1–35, 1998.
[35] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, 1996.
[36] Z. Wang, P. Song, and M. Pauly, "State of the art on computational design of assemblies with rigid parts," *Comput. Graph. Forum*, vol. 40, no. 2, pp. 633–657, 2021.
[37] J. Xiao, "Automatic determination of topological contacts in the presence of sensing uncertainties," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1993, pp. 65–70.
[38] X. Ji and J. Xiao, "Planning motions compliant to complex contact states," *Int. J. Robot. Res.*, vol. 20, no. 6, pp. 446–465, 2001.
[39] C. Ericson, *Real-Time Collision Detection*. Boca Raton, FL, USA: CRC Press, 2004.
[40] F. Wang and K. Hauser, "Dense robotic packing of irregular and novel 3D objects," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 1160–1173, Apr. 2022.
[41] K. Goldberg, B. V. Mirtich, Y. Zhuang, J. Craig, B. R. Carlisle, and J. Canny, "Part pose statistics: Estimators and experiments," *IEEE Trans. Robot. Automat.*, vol. 15, no. 5, pp. 849–857, Oct. 1999.
[42] Y. Maeda, Y. Goto, and S. Makita, "A new formulation for indeterminate contact forces in rigid-body statics," in *Proc. 2009 IEEE Int. Symp. Assem. Manuf.*, 2009, pp. 298–303.
[43] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, vol. 2, pp. 995–1001.
[44] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proc. Conf. Comput. Graph. Interactive Techn.*, 1997, pp. 209–216.

Philippe Nadeau received the B.Eng degree, with honours, in automated manufacturing engineering from École de Technologie Supérieure, Montréal, Canada. He is currently working toward the Ph.D. degree with the STARS laboratory, University of Toronto Institute for Aerospace Studies.

His doctoral work focused on designing novel algorithms to enhance collaborative robots autonomy in object handling tasks. His research interests lie at the intersection of perception and planning with an emphasis on fast and general algorithms deployed on real robots in human-centric environments.

He was a recipient of various awards for his research work, including Canada's Graduate Scholarships, Québec's Master 's Research Scholarship, and Vector Institute's Scholarship in Artificial Intelligence.

Jonathan Kelly (Senior Member, IEEE) received the Ph.D. degree from the University of Southern California, California, LA, USA, in 2011. His doctoral work focused on temporal and spatial calibration for high accuracy visual-inertial motion estimation.

From 2011 to 2013, he was a Postdoctoral Fellow with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA. He is currently an Associate Professor and Director with the Space and Terrestrial Autonomous Robotic Systems (STARS) Laboratory, University of Toronto Institute for Aerospace Studies, Toronto, Canada. His research interests include perception, planning, and learning for interactive robotic systems.

Dr. Kelly holds the Canada Research Chair in Collaborative Robotics.