

Stochastic polycube models for robotic grasping research

Miles Hansard

Queen Mary University of London

miles.hansard@qmul.ac.uk

November 2025

Abstract

This paper introduces a family of random 3D shapes, based on polycube constructions. Polycubes are face-connected assemblies of identical cubes, which can generate a wide variety of shapes, including those with concavities and holes. It is argued that polycube datasets are very suitable for robotic grasping research, and related fields. A suitable shape generation algorithm is proposed, which can preferentially generate flatter or more elongated configurations, represented by triangular meshes. It is also shown that the local surface geometry can be varied, by random perturbation of the mesh vertices, while preventing self-intersections. This constraint allows analytic expressions to be obtained for the volume, surface area, and inertia tensor, in all cases. The conversion of the meshes to point clouds is also described, so that the resulting models resemble the data obtained from depth cameras and lidar sensors. An efficient implementation is provided, as a Julia package.

1 Introduction

Simulation-based training has proved to be very effective for robotic grasping and manipulation systems [Newbury et al., 2023]. Datasets of 3D object scans, typically in the form of point clouds, are widely used in this context [Mahler et al., 2017, Fang et al., 2020]. The construction of these datasets often involves a somewhat subjective choice of target objects, such as the YCB collection, in conjunction with a set of reference grasps. More generally, these methods require a database of objects that either matches or encompasses the real world distribution, in order to facilitate transfer learning. A matching distribution can be approximated by collecting scans of prototypical objects, such as those in the YCB dataset [Calli et al., 2017, Fang et al., 2020], thereby reducing the domain gap [Bousmalis et al., 2018, Ma et al., 2024]. Alternatively, an encompassing distribution can be obtained by synthesizing random shapes, thereby achieving domain randomization [Tobin et al., 2018]. Focusing

on domain randomization, it would be useful to have a procedure to generate:

- A variety of random shapes; including convex, non-convex, regular, and irregular instances.
- Shapes that resemble manufactured objects, e.g. boxes or industrial parts.
- Shapes for which the volume, surface area, and other geometric properties can be computed (and differentiated) analytically.

It is difficult to achieve these objectives, firstly because the appropriate definition of a ‘random’ 3D shape is unclear, in the context of robotic grasping. In particular, it is desirable to parameterize global shape variation (in relation to grasp stability), as well as local surface variation (in relation to contact friction). The obvious approaches (such as taking the α -hull [Edelsbrunner and Mücke, 1994] of a random point cloud) tend to produce highly irregular shapes, which do not resemble anything that might be encountered in a typical domestic or industrial environment.

A parametric family of random *polycube* [Gardner, 1972] shapes is proposed here, in sections 2–3. These models, which are face-connected assemblies of cubes, encompass a wide variety of global shapes, and resemble generic manufactured objects. In order to incorporate local surface variation, the polycubes are converted to triangular meshes, and the vertices are subject to controlled 3D perturbations. The proposed shape family is designed to be quite generic, for use in domain-randomized grasp training, according to the three criteria listed above.

It will usually be necessary for a robotic system to *manipulate* the target object, having successfully grasped it, in order to perform a given task. The dynamics of the manipulation task will depend on the inertial properties of the object, in simulated or actual execution [Mavrakis and Stolkin, 2020]. In particular, the object’s volume (or mass) and inertia tensor are important [Brook et al., 1998], [Harwin and Barrow, 2013], [Lippiello et al., 2013]. It will be shown in sections 3.3 that these properties of a deformed polycube object can be evaluated analytically, as functions of the vertex coordinates. These comput-

tations are greatly facilitated by the avoidance of mesh self-intersections, during the deformation procedure.

1.1 Related work

There is relatively little work on the *generation* of object models for robotic grasping, as addressed in this report. Pokorny et al. [2013] approached the problem by using random spherical harmonic expansions, which tend to produce smooth and axis-symmetric objects. In contrast, Morrison et al. [2020] used evolutionary optimization to synthesize a mesh dataset, subject to shape complexity and diversity constraints. Tobin et al. [2018] also produced a mesh dataset, by randomly combining objects from an existing CAD database. Andries et al. [2025] proposed a hybrid approach, in which a variational auto-encoder is trained on an existing shape database, and then used as a generative model. Wang et al. [2019] show how to deform a given mesh, in order to produce a more adversarial object, with respect to a measure of grasp quality.

Several different *representations* of 3D objects have been used for robotic grasping research. These include polyhedral meshes [Ponce et al., 1997], depth maps [Jiang et al., 2011], bounding boxes [Huebner et al., 2008], superquadrics [Vezzani et al., 2017], spherical harmonic expansions [Pokorny et al., 2013], and point clouds [Ni et al., 2020]. The last of these has become ubiquitous, owing to the commoditization of RGBD cameras and other depth sensors [Fang et al., 2020].

Polycube models have been studied in many different contexts, including statistical physics (as *lattice animals*, Harary 1967), group theory [Lunnon, 1972], topology (as cubical *complexes*, Manin et al. 2020), combinatorics [Aleksandrowicz and Barequet, 2006], biology (as the *Eden model*, Eden 1961, computer graphics [Huang et al., 2014], and recreational mathematics (as *soma pieces*, Gardner 1986, Berlekamp et al. 2004). Images of polycubes have also been used as visual stimuli in cognitive science [Shepard and Metzler, 1971], and as training data, in computational rendering [Eslami et al., 2018]. Physical polycubes, made of brass and wood, have already been used in human grasping and manipulation experiments [Klein et al., 2020].

1.2 Structure and contributions

Section 2 contains the main practical contribution of the paper, which is a general procedure for generating random polycube models (2.1), including novel schemes for controlling the overall shape (2.2), and surface irregularity (2.3), while ensuring that no self-intersections can occur. The conversion to point cloud form is also described (2.4). Section 3 applies standard methods to obtain analytic expressions for the mass and centroid (3.1), mass covariance matrix (3.2), and inertia tensor (3.3) of any polycube model. The discrete curvature of the per-

turbed surfaces is also defined (3.4). It is shown that the mass covariance can also be used to define useful measure of distance between 3D shapes (3.5). Section 4 shows how the preceding concepts can be applied in a subtask of grasp planning: the placement of contacts in relation to surface curvature, such as roofs and valleys. Further potential applications are discussed in section 5.

A complete implementation is provided in the associated Julia package, which was used to perform all computations described below:

<https://github.com/mhansard/PolycubeModels.jl>

1.3 Notation

The short notation $1:N$ will be used for the index range $1, 2, \dots, N$. Vectors and matrices will be written in bold type, as \mathbf{v} and \mathbf{M} , respectively. Points will be represented by column vectors, while $\mathbf{0}_N$ and $\mathbf{1}_N$ will denote N -element column vectors of zeros or ones, respectively. It will be useful to recall the *Minkowski sum* operation:

$$\mathbf{x} \oplus \mathcal{D} := \{\mathbf{x} + \mathbf{d} : \mathbf{d} \in \mathcal{D}\} \quad (1)$$

which performs the geometric ‘dilation’ of the point \mathbf{x} by all vectors in the set \mathcal{D} . In particular, this operation will be used in conjunction with the following notation for sets of 3D lattice vectors:

$$\mathcal{D}_R := \{\mathbf{d} : d_i \in \{0, \pm 1\}, |\mathbf{d}|_1 = R\} \quad (2)$$

where $\mathbf{d} = [d_1, d_2, d_3]^\top$, and the positive integer R is the L_1 ‘radius’ of the set. For example, \mathcal{D}_1 comprises the vectors $[\pm 1, 0, 0]^\top$, $[0, \pm 1, 0]^\top$ and $[0, 0, \pm 1]^\top$, so the set $\mathbf{x} \oplus \mathcal{D}_1$ contains the six neighbours of point \mathbf{x} on the unit cubic lattice. It will also be useful, in section 2.2, to define unions of these sets, with the notation $\mathcal{D}_{R,S} := \mathcal{D}_R \cup \mathcal{D}_S$.

2 Polycube object models

This section describes the representation of a solid polycube object, as a simplicial complex, in which each cube is composed of twelve tetrahedra. A convenient face-vertex notation is established, in order to facilitate the derivations in sections 2.2 and 2.3, as well as the computational implementations.

2.1 Geometric structure

A polycube object \mathcal{P} is the union of $m = 1:M$ face-connected unit cubes. Each cube will be centred on an integer lattice point \mathbf{c}_m , with the eight surrounding vertices given by

$$\bar{\mathcal{X}}_m = \mathbf{c}_m \oplus \frac{1}{2}\mathcal{D}_3 \quad (3)$$

where \mathcal{D}_3 is the set of vectors $[\pm 1, \pm 1, \pm 1]^\top$. The overbar indicates that $\bar{\mathcal{X}}_m$ contains the original, unperturbed vertices. In order to build the mesh representation, each

M	1	2	3	4	5	6	7	8
T_M^*	1	1	2	8	29	166	1023	6922
\tilde{T}_M^*	1	1	2	7	23	112	607	3811
T_M	1	1	2	8	28	150	852	5299
\tilde{T}_M	1	1	2	7	22	103	515	2976

Table 1: Total numbers T_M^* of distinct ‘oriented’ polycube shapes [OEIS A000162], of orders $M = 1:8$, followed by total numbers \tilde{T}_M^* of distinct ‘unoriented’ shapes, after identification of any reflected instances [OEIS A038119]. The present ensembles further exclude any shapes with exposed edge or vertex contacts, resulting in total numbers T_M and \tilde{T}_M of oriented and unoriented shapes, respectively. The basic test set of 11 shapes in Fig. 1 comprises the union of the bold instances in the last row.

distinct face in the union of cubes \mathcal{P} will be randomly split across one of the two diagonals. Each resulting triangle on the external surface of \mathcal{P} forms a *single* simplex, when associated with the corresponding cube centre c_m . Similarly, each triangle on an internal face of \mathcal{P} forms a *pair* of simplices, when associated with the centres of the adjoining cubes.

It is necessary to merge the duplicate vertices, which arise from adjoining the cubes, so that the final mesh can be deformed without tearing. Hence let $\mathbf{x}[\cdot]$ be a list of all *distinct* cube vertices of the complete object \mathcal{P} . Then let $\{\ell_n^1, \ell_n^2, \ell_n^3\}$ be a mapping from any simplex index n to the three associated vertices in the indexed list $\mathbf{x}[\ell]$. Similarly, let $\mathbf{c}[k]$ be the list of cube centres, so that $\mathbf{c}[k_n]$ is associated with the n -th simplex. The complete solid polycube \mathcal{P}_M , comprising M cubes, is represented by $N = 12M$ distinct simplices \mathcal{S}_n . Each simplex is represented by a matrix of four column vectors, in three dimensions:

$$\mathbf{S}_n = \left[\mathbf{c}[k_n], \mathbf{x}[\ell_n^1], \mathbf{x}[\ell_n^2], \mathbf{x}[\ell_n^3] \right]_{3 \times 4} \quad (4)$$

where $n = 1:N$.

Having established the construction, above, it will be more convenient to suppress the indices, and to partition a given simplex matrix \mathbf{S} , as follows:

$$\mathbf{S} = [\mathbf{c} | \mathbf{X}] \quad \text{where } \mathbf{X} = [\mathbf{x}_1 | \mathbf{x}_2 | \mathbf{x}_3] \quad (5)$$

Finally, the vertices (but not the centres) of the polycube will be subject to random perturbations around their initial values \mathbf{x} . More precisely, if $\delta(\sigma)$ is a random vector, drawn from a distribution with parameter σ , then

$$\mathbf{x} = \bar{\mathbf{x}} + \delta(\sigma), \quad \text{where } |\delta(\sigma)|_2 < \delta_{\max} \quad (6)$$

with $\bar{\mathbf{x}}$ being the original vertices (3) from the sets $\bar{\mathcal{X}}_m$. Note that each random vertex \mathbf{x} is confined to a sphere of radius δ_{\max} around $\bar{\mathbf{x}}(\ell)$. This prevents the vertex perturbations from causing any intersections between the triangular faces. The probability density of $\delta(\sigma)$ will be defined in section 2.3.

2.2 Shape generation

It can be seen from Table 1 that the total number T_M^* of distinct polycubes grows very rapidly with respect to the number M of cubes involved; indeed the count is known only up to T_{22}^* (being over 10^{16}) at present [OEIS A000162]. The table also gives the counts \tilde{T}_M of the sets $\{\mathcal{P}_M\}$ used here, after exclusion of reflected and degenerate objects, as described in section 2.2. In particular, it is convenient to define a canonical **test set** as the union $\{\mathcal{P}_M\}$ for $M = 1:4$. This gives 10 distinct shapes, plus the original unit cube, as shown in Fig. 1.

For the construction of larger datasets $\{\mathcal{P}_M\}$, containing T polycube shapes, two problems arise. Firstly, if $T < T_M^*$, then which of the possible shapes should be chosen? Secondly, if $T > T_M^*$, then which of the possible shapes should be multiply included? These problems are significant, because T and T_M^* could be *very* different, in practice. A simple solution is proposed here, in the form of an algorithm that randomly ‘grows’ a polycube, up to the required size M .

Let $\mathcal{C} = \{c_k\}$ be an existing set of one or more cube centres, which are positioned on the standard integer lattice. The face-connected boundary sites \mathcal{B} can be obtained by adding each vector in \mathcal{D}_1 to each point in \mathcal{C} , and then removing all of the original points from the resulting dilation, $\mathcal{C} \oplus \mathcal{D}_1$. A new face-connected centre can then be generated by taking a random sample \mathbf{b} from the boundary sites in \mathcal{B} . More formally, M cube centres can be generated by initializing $\mathcal{C} = \{0_3\}$, and then performing the following iteration, $M - 1$ times:

$$\begin{aligned} \mathcal{B} &\leftarrow (\mathcal{C} \oplus \mathcal{D}_1) \setminus \mathcal{C} \\ \mathcal{C} &\leftarrow \text{sample}(\mathcal{B}) \cup \mathcal{C} \end{aligned} \quad (7)$$

Although this procedure always generates face-connected objects, it does not prevent the structure from folding-back, and making *additional* non-face contacts with itself at exposed edges or vertices. Indeed, there are many such cases in the original set of polycubes, as can be seen by comparing the upper and lower sections of Table 1. It is better to exclude these cases, so that the polycube resembles a typical manufactured object. Furthermore, this restriction also ensures that basic geometric properties, such as the discrete curvature 23 are well-defined. A suitable test is described below, in a way that can easily be implemented, so that rejection sampling can be used in (7).

Suppose that \mathbf{b} is a candidate site, generated by the sampling procedure described above, so that the corresponding cube would be face-connected to the existing set \mathcal{C} . The complete system of contacts *around* \mathbf{b} can be defined as

$$\mathcal{C}_1(\mathbf{b}) = \mathcal{C} \cap (\mathbf{b} \oplus \mathcal{D}_1) \quad (8)$$

$$\mathcal{C}_{2,3}(\mathbf{b}) = \mathcal{C} \cap (\mathbf{b} \oplus \mathcal{D}_{2,3}) \quad (9)$$

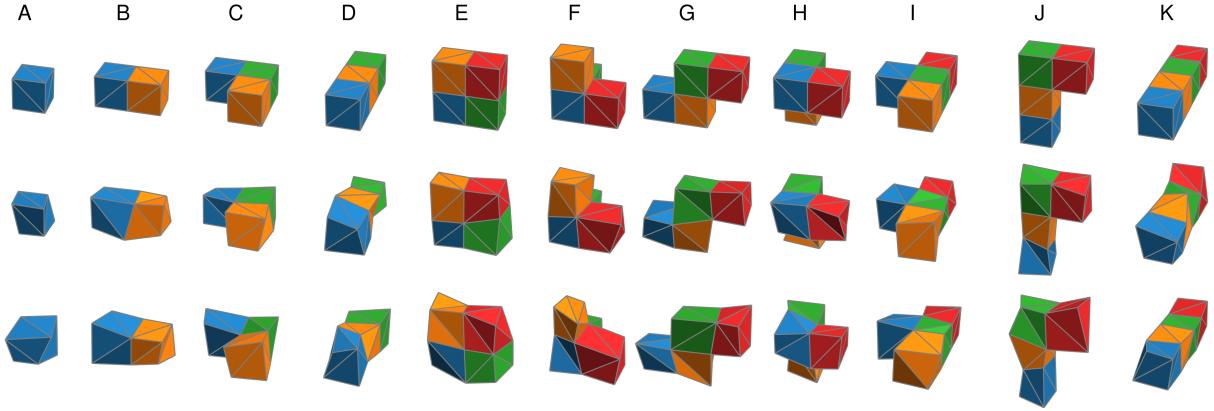


Figure 1: Top: all eleven polycubes of orders $M = 1, 2, 3, 4$, labelled A, B, C–D, and E–K, respectively. One reflection (of H) is excluded from the complete set, as enumerated in table 1. The labels A–K are assigned in order of shape distance (26) from the unit cube A. Note that the partition of each square face into two triangles is performed randomly. Middle, bottom: the same shapes after random deformation, with $\alpha = 0.25$ and $\alpha = 0.5$ respectively.

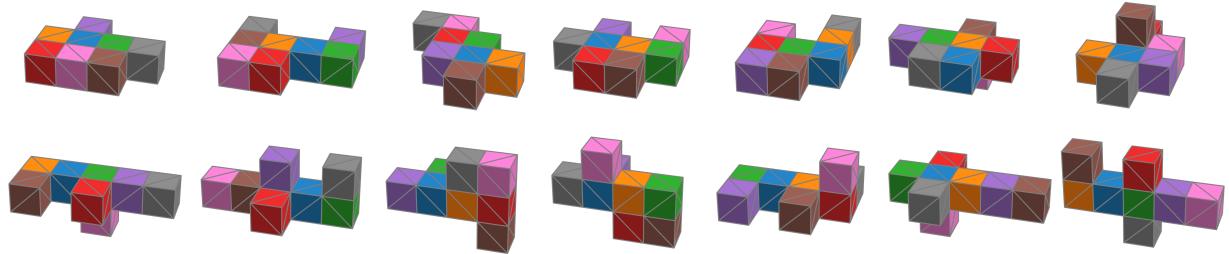


Figure 2: Random polycubes of order 8, biased towards flatter shapes (top row), or longer shapes (bottom row). The semi-axes of the Gaussian shape field (11) are $[2, 2, \sigma]$ for the flat objects, and $[4, \sqrt{\sigma}, \sqrt{\sigma}]$ for the long objects, with $\sigma = 1/3$. Note that the volumes of the two shape ellipsoids, being the products of the semi-axes, are equal.

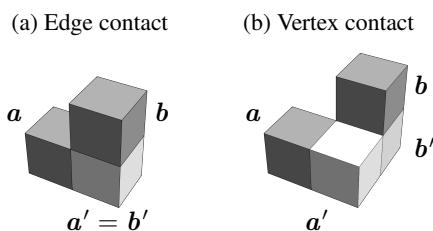


Figure 3: Suppose that the two dark grey cubes a and b are part of a larger face-connected complex (not shown). Edge-contact (a) between a and b is acceptable if supported by a common neighbour $a' = b'$. Vertex-contact (b) is acceptable if supported by neighbours a' and b' , provided the latter are themselves adjacent.

Here $\mathcal{C}_1(\mathbf{b})$ is the set of face contacts with the original polycube \mathcal{C} , which is non-empty by construction. In addition, the (possibly empty) set $\mathcal{C}_{2,3}(\mathbf{b})$ contains the potentially ‘bad’ edge or vertex contacts with the original polycube, if any. Specifically: $\mathcal{D}_{2,3}$ is the union of the 12 possible edge contacts from \mathcal{D}_2 with the eight possible vertex contacts from \mathcal{D}_3 . The candidate site \mathbf{b} can be accepted if all non-face contacts in $\mathcal{C}_{2,3}(\mathbf{b})$ are face-connected via another cube or pair of cubes, as shown in

Fig. 3. More formally, proposal \mathbf{b} is accepted if the set of bad contacts $\mathcal{C}_{2,3}(\mathbf{b})$ is empty, or if:

$$\min_{\mathbf{a}' \in \mathcal{C}_1(\mathbf{a}), \mathbf{b}' \in \mathcal{C}_1(\mathbf{b})} |\mathbf{a}' - \mathbf{b}'|_1 \leq 1, \quad \text{for all } \mathbf{a} \in \mathcal{C}_{2,3}(\mathbf{b}). \quad (10)$$

Note that there are two qualitative scenarios in which a bad contact is accepted, as illustrated below. If $\min|\mathbf{a}' - \mathbf{b}'|_1 = 0$, then the edge contact between \mathbf{a} and \mathbf{b} is supported by at least one face-connected chain, $[\mathbf{a}, \mathbf{a}' = \mathbf{b}', \mathbf{b}]$. Alternatively, if $\min|\mathbf{a}' - \mathbf{b}'|_1 = 1$, then the vertex contact between \mathbf{a} and \mathbf{b} is supported by at least one face-connected chain $[\mathbf{a}, \mathbf{a}', \mathbf{b}', \mathbf{b}]$, with $\mathbf{a}' \neq \mathbf{b}'$. Conversely, if $\min|\mathbf{a}' - \mathbf{b}'|_1 \geq 2$, then the proposal \mathbf{b} is rejected.

The sampling procedure in (7) is uniform over the boundary sites \mathcal{B} of the object. It is straightforward to bias the procedure towards a given overall distribution of mass, by introducing a reference covariance matrix Σ , so that the sampling weights in (7) can be defined by the zero-mean Gaussian density

$$P_{\mathcal{B}}(\mathbf{b} | \Sigma) \propto \exp(-\frac{1}{2} \mathbf{b}^\top \Sigma^{-1} \mathbf{b}) \quad \text{where } \mathbf{b} \in \mathcal{B}. \quad (11)$$

Suppose (without loss of generality) that Σ is diagonal, with variances $[\sigma_1^2, \sigma_2^2, \sigma_3^2]$. If the three variances are

equal, then lower settings of the common parameter will tend to produce more compact isotropic objects. Alternatively, if $\sigma_1^2 = \sigma_2^2$, while σ_3^2 is lower, then flattened objects are more likely. Finally, if $\sigma_2^2 = \sigma_3^2$, while σ_1^2 is much higher, then elongated objects are more likely, as shown in Fig. 4. The global size, shape, and orientation of the object can be further modified by applying any non-degenerate 3D affine transformation to the vertices.

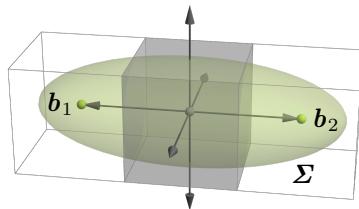


Figure 4: The initial shape (grey cube) can be extended to any of the six sites in the face-connected boundary \mathcal{B} (arrows). Sites b_1 and b_2 are more likely to be added, due to the influence of the shape-field Σ , from (11), which favours elongated objects in this example.

The above procedures can be used to generate random shape ensembles, which will typically contain duplicates and reflections. It may be desirable, in some applications, to identify (and perhaps remove) duplicate and/or reflected shapes. This task is complicated by the fact that the site coordinates may be different; e.g. there are six versions of the simple two-cube shape. To resolve this, it is straightforward to first generate the 24 special orthogonal 3×3 matrices that represent all discrete rotations in the cubic lattice. In order to determine whether or not two shapes \mathcal{P} and \mathcal{P}' are identical up to rotation, begin by centring their site coordinates (so that each shape has zero mean), and then apply the full set of matrices to each set of sites. This generates the ‘orbit’ of each shape, with respect to the rotation group. The two orbits can each be sorted lexicographically (by their concatenated site coordinates), and then compared, or given a unique hash identifier for later use. Note that this procedure does not require any shape ‘correspondence’ to be established, and can be performed entirely by integer operations. The procedure can also incorporate reflections, using the larger discrete group of 48 orthogonal matrices.

2.3 Surface deformation

Each vertex x is to be perturbed around the lattice point \bar{x} , by a random vector $\delta(\sigma)$, as in (6). The distribution of x should be spherically symmetric, and bounded by a maximum radius δ_{\max} to ensure that no self-intersections of the mesh can occur. It is convenient to analyze the case of a simplex $[\mathbf{0}, \mathbf{X}]$, located at the origin, without loss of generality. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$, as before, and suppose that the edge $[\mathbf{x}_1, \mathbf{x}_2]$ cuts diagonally across the square face that contains the three vertices. The radius δ_{\max} should be as large as possible, while ensuring that no set of perturbations can cause the simplex to collapse. Equivalently,

if each vertex is surrounded by a sphere of radius δ_{\max} , then there is a bisecting plane, which passes through the origin, and is tangent to all three spheres. If the perturbations are strictly inside the corresponding spheres, then the simplex cannot collapse into the bisecting plane, as shown in Fig. 5.

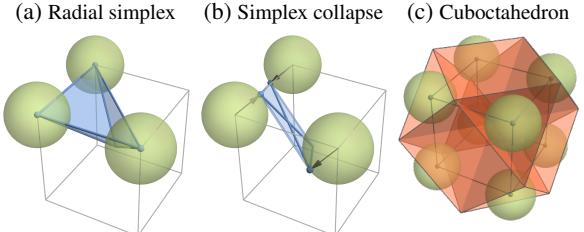


Figure 5: Each cube comprises 12 radial simplices, one of which is shown in blue (a). The mesh topology is stable with respect to perturbation of the vertices inside the three spheres. The complete collapse of a simplex to a limiting plane, shown in (b), is forbidden. The full system of constraints defines a cuboctahedron (c), in which the limiting planes (red) are wedged between the perturbation spheres at the eight vertices of the cube.

It remains to determine the plane, so that the common sphere radius can be computed. The normal vector \mathbf{n} must lie in the perpendicular plane spanned by $\mathbf{x}_1 + \mathbf{x}_2$ and \mathbf{x}_3 , i.e. it must be parallel to $\mathbf{m} = \gamma(\mathbf{x}_1 + \mathbf{x}_2) + \mathbf{x}_3$, for some value of γ . Setting the distances from the plane to be equal, on either side $-\mathbf{x}_k \cdot \mathbf{m} = \mathbf{x}_3 \cdot \mathbf{m}$ with $k = 1$ or $k = 2$, leads to the solution $\gamma = -1$. The unit normal can then be computed from $\mathbf{n} = \mathbf{m}/|\mathbf{m}|$, and the distances from the original vertices to the separating plane are found to be

$$\delta_{\max} = |\mathbf{x}_k \cdot \mathbf{n}| = \frac{1}{2\sqrt{3}} \quad \text{for } k = 1:3. \quad (12)$$

There are $\binom{4}{3} = 4$ bisecting planes in the complete configuration, which can be defined from all distinct triplets, taken from the four vertices of any square face. Geometrically, these planes define a *cuboctahedron*, when intersected with the original cube, as shown in Fig 5c.

Having determined the maximum radius, the perturbation $\delta(\sigma)$ can be sampled from a spherically symmetric Normal distribution, truncated at distance δ_{\max} . It remains to determine a suitable value for σ , the standard deviation of the distribution. This can be done by positioning the mode of the *radial* distribution suitably inside the radius δ_{\max} . The squared length $|\delta(\sigma)|^2$ of the three-dimensional normally distributed vector $\delta(\sigma)$ is distributed as $\text{Gamma}(3/2, 2\sigma^2)$. From this, the distribution of the length itself can be obtained, which is the generalized gamma density $\text{Gamma}(3/2, \sigma\sqrt{2}, 2)$, the mode being at $\sigma\sqrt{2}$. This suggests a parameterization of σ that interpolates between a point mass at \bar{x} , and a distribution with radial maximum at δ_{\max} , as follows:

$$\sigma_\alpha = \frac{\alpha \delta_{\max}}{\sqrt{2}} = \frac{\alpha}{2\sqrt{6}}, \quad \text{where } \alpha \in [0, 1]. \quad (13)$$

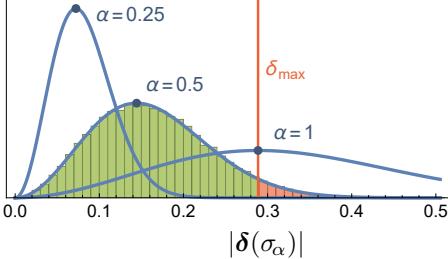


Figure 6: Example generalized Gamma densities (blue) of the perturbation magnitudes $|\delta(\sigma_\alpha)|$ that arise from spherical Normal(σ_α) vertex perturbations. The parameterization (13) ensures that the mode (blue dot) does not exceed the geometric self-intersection limit (12). Rejection sampling is used to eliminate any excessive perturbations, e.g. those shown in red, for the default setting $\alpha = \frac{1}{2}$.

A suitable default choice is $\alpha = \frac{1}{2}$, which leaves less than 5% of the full density outside the radius δ_{\max} , as shown in Fig. 6. Rejection sampling can be used to generate random Normally distributed vectors $\delta(\sigma_\alpha)$, retaining only those samples for which $|\delta(\sigma_\alpha)| < \delta_{\max}$. The advantage of having parameterized the mode of the radial distribution is that, unlike the mean, it is unaffected by the rejection process.

2.4 Point cloud form

It is necessary to sample the surface of the polycube, in order to simulate a point cloud scan, and/or to generate possible grasp contacts. The latter task will be considered in section 4, based on the default sampling model that is developed here. It is straightforward to sample uniform points from a triangle [Turk, 1990], but it is also well known that the resulting distribution will be uneven; there will be gaps and clusters, neither of which is desirable. Rather than ‘thinning’ a dense but uneven distribution [Corsini et al., 2012], the planar *Poisson disk* method of Bridson [2007] will be adapted to the meshed surface. This ensures that no sample can be generated within a given radius ρ_{\min} of an existing sample. The algorithm maintains a list of active sites; at a randomly chosen site, candidate samples are generated in the surrounding annulus of radii $[\rho_{\min}, 2\rho_{\min}]$. If one of these samples is further than ρ_{\min} from all existing samples, then it is accepted, and also becomes a new site. Conversely, if none of the candidate samples are acceptable, then the current site is marked inactive. This process continues until the active list becomes empty.

A 3D occupancy grid (implemented as a hash table), of cell size $\rho_{\min}\sqrt{3}$, is used for the sample queries. Note that the algorithm cannot be decomposed into separate 2D procedures for each face, because this would lead to oversampling around the seams. Instead, co-planar samples are generated in an orthonormal basis for each face $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$, and then tested against the shared occupancy grid in the ambient space. In particular, let

$$\mathbf{E} = [\mathbf{x}_2 - \mathbf{x}_1 \mid \mathbf{x}_3 - \mathbf{x}_1] \quad (14)$$

be the edge vectors of a given triangle. Given the singular value decomposition of this matrix, the columns of the first factor are an orthonormal basis for the corresponding plane in the ambient space. Furthermore, it is essential to discard any sample \mathbf{x} that falls outside the current triangle (and potentially off the surface). This can be done by checking that the barycentric coordinates β of \mathbf{x} are non-negative, where $\mathbf{x} = \mathbf{X}\beta$. These coordinates are obtained as

$$\beta = [1 - |\alpha|_1, \alpha]^\top \quad \text{where } \alpha = \mathbf{E}^+(\mathbf{x} - \mathbf{x}_1). \quad (15)$$

The 2×3 matrix \mathbf{E}^+ is the pseudoinverse of (14), which is readily obtained from the SVD that was used to construct the orthonormal basis, above.

3 Polycube properties

This section gives analytic definitions for the mass, centroid, scatter matrix, and inertia tensor of any polycube object. These properties are useful for dataset design, as well as for grasp planning and performance analysis.

3.1 Mass and centroid

The object centroid is an important property for grasp planning. For example, if a parallel gripper has two contact points on the object, then the axis between the contacts should pass close the object centroid, in order to minimize gravitational and inertial effects.

If the object \mathcal{P} has uniform density, as will be assumed throughout, then the overall centroid $\bar{\mathbf{s}}_{\mathcal{P}}$ will be the sum over all simplex centroids $\bar{\mathbf{s}}_i$, weighted by the corresponding volumes $|\mathcal{S}_i|$. The definitions are straightforward, even after deformation of the constituent cubes, because there cannot be any intersections between the simplices in \mathcal{P} :

$$\begin{aligned} \bar{\mathbf{s}}_i &= \frac{1}{4} \mathbf{S}_i \mathbf{1}_4 & |\mathcal{S}_i| &= \frac{1}{6} |\det[\mathbf{S}_i^\top \mid \mathbf{1}_4]| \\ \bar{\mathbf{p}} &= \sum_{i=1}^N \frac{|\mathcal{S}_i|}{|\mathcal{P}|} \bar{\mathbf{s}}_i & |\mathcal{P}| &= \sum_{i=1}^N |\mathcal{S}_i| \end{aligned} \quad (16)$$

Note that if the vertex perturbations (6) are small, then the total volume is $|\mathcal{P}| \approx M$, and the overall centroid is $\bar{\mathbf{p}} \approx \frac{1}{M} \sum_k^M \mathbf{c}_k$.

The surface area $A(\mathcal{P})$ is easily obtained as the sum of the external triangle areas. It will also be useful to define the isoperimetric ratio [Berger, 2010], as a measure of compactness:

$$C(\mathcal{O}) = 6^3 |\mathcal{O}|^2 / A^3(\mathcal{O}) \quad (17)$$

where $|\mathcal{O}|$ and $A(\mathcal{O})$ are the volume and surface area of shape \mathcal{O} , respectively. The 6^3 scaling performs the normalization $C(\mathcal{P}_1) = 1$ with respect to the unit cube; less compact shapes have lower ratios, as shown in Table 2, with reference to Fig. 1.

3.2 Covariance

It will be useful to obtain analytic expressions for the covariance matrix \mathbf{Q} of a polycube, considered as a uniform solid body, for three reasons. Firstly, the eigenvectors of \mathbf{Q} correspond to the principle axes of the shape, while the corresponding eigenvalues characterize the global shape. Secondly, the current covariance matrix can be used to influence the location of additional cubes, so that a given overall shape can be approximated, as described in section 2.2. Thirdly, the inertia tensor of the polycube, which is crucial to dynamic manipulation, can easily be obtained from the scatter matrix, as described below.

Consider the unit K -dimensional simplex \mathcal{R} , with associated vertices $[\mathbf{0}_K, \mathbf{I}_K]$. It will be convenient to use angle-brackets to denote the integral over this reference shape, e.g. $\langle \mathbf{r} \rangle := \int_{\mathcal{R}} \mathbf{r} d\mathbf{r}$. Then, using the Stroud formula [Grundmann and Möller, 1978] for the integral of a monomial over the reference simplex:

$$\langle \mathbf{r} \rangle = \int_{\mathcal{R}} \mathbf{r} d\mathbf{r} = \frac{\mathbf{1}_K}{(K+1)!} \quad (18)$$

$$\langle \mathbf{r} \mathbf{r}^\top \rangle = \int_{\mathcal{R}} \mathbf{r} \mathbf{r}^\top d\mathbf{r} = \frac{\mathbf{I}_K + \mathbf{1}_K \mathbf{1}_K^\top}{(K+2)!}. \quad (19)$$

The denominators are functions of the simplex dimension, with $K = 3$ in this case, hence $\langle \mathbf{r} \rangle = \mathbf{1}_3/24$, while $\langle \mathbf{r} \mathbf{r}^\top \rangle_{ij} = 1/60$ on the diagonal $i = j$, and $1/120$ elsewhere. Also note that the centroid of \mathcal{R} can be expressed as $\langle \mathbf{r} \rangle / |\mathcal{R}|$, where $|\mathcal{R}| = 1/6$ is the volume of the reference simplex (as the unit cube accommodates $1/K!$ unit simplices).

Without loss of generality, any simplex $\mathcal{S} = [\mathbf{c}, \mathbf{X}]$ can be translated to the origin, so that the associated matrix becomes $[\mathbf{0}, \mathbf{D}]$, where $\mathbf{d}_j = \mathbf{x}_j - \mathbf{c}$ are the columns of \mathbf{D} . Then the coordinates of any point \mathbf{x} in the translated volume are linear functions $\mathbf{x} = \mathbf{D}\mathbf{r}$ of the reference simplex coordinates, with associated Jacobian $\nabla \mathbf{x}(\mathbf{r}) = \mathbf{D}$. The general scatter matrix \mathbf{Q} of \mathcal{S} can now be evaluated, by change of variables, as follows:

$$\begin{aligned} \mathbf{Q}(\mathcal{S}) &= \int_{\mathcal{S}} (\mathbf{s} - \bar{\mathbf{s}})(\mathbf{s} - \bar{\mathbf{s}})^\top d\mathbf{s} \\ &= |\det(\mathbf{D})| \int_{\mathcal{R}} (\mathbf{D}\mathbf{r} - \bar{\mathbf{s}})(\mathbf{D}\mathbf{r} - \bar{\mathbf{s}})^\top d\mathbf{r} \quad (20) \\ &= \frac{|\mathcal{S}|}{|\mathcal{R}|} \mathbf{D} \left(\langle \mathbf{r} \mathbf{r}^\top \rangle - \frac{\langle \mathbf{r} \rangle \langle \mathbf{r} \rangle^\top}{|\mathcal{R}|} \right) \mathbf{D}^\top \end{aligned}$$

where $\bar{\mathbf{s}} = \mathbf{D} \langle \mathbf{r} \rangle / |\mathcal{R}|$ is the centroid of \mathcal{S} , and $|\mathcal{S}|/|\mathcal{R}| = |\det(\mathbf{D})|$ is the change of volume due to \mathbf{D} . The derivation of (20) involves the observation that, after expanding the quadratic form, the constant term is multiplied by the volume of the integration region, $\int_{\mathcal{R}} \bar{\mathbf{s}} \bar{\mathbf{s}}^\top d\mathbf{r} = |\mathcal{R}| \bar{\mathbf{s}} \bar{\mathbf{s}}^\top$. This fact can be used again, in defining the combination of multiple covariance matrices, as follows. Let \mathbf{Q} be the overall scatter matrix of the polycube shape \mathcal{P} , and let $\mathbf{Q}(\mathcal{S}_i)$ be the scatter matrices of the $i = 1:N$

constituent simplices, computed from (20). The matrices $\mathbf{Q}(\mathcal{S}_i)$ can be expressed in relation to the overall centroid, and then summed:

$$\mathbf{Q}_{\mathcal{P}} = \sum_i^N \mathbf{Q}(\mathcal{S}_i) + |\mathcal{S}_i| \mathbf{t}_i \mathbf{t}_i^\top, \quad \mathbf{t}_i = \bar{\mathbf{s}}_i - \bar{\mathbf{p}}. \quad (21)$$

In particular, if the polycube is shifted so that $\bar{\mathbf{p}} = \mathbf{0}_3$, then it can be seen that each simplex \mathcal{S}_i contributes a scatter matrix $\mathbf{Q}(\mathcal{S}_i)$, plus a ‘point moment’ associated with its centroid $\bar{\mathbf{s}}_i$.

3.3 Inertia tensor

The inertia tensor \mathbf{M} is a symmetric linear map [Arnold, 1989] from the angular velocity vector $\boldsymbol{\omega}$ of a rigid body to the resulting angular momentum vector, $\mathbf{m} = \mathbf{M}\boldsymbol{\omega}$. If \mathbf{M} is defined at the centroid of the object, then the time derivative $\dot{\mathbf{m}}$ of gives the total torque, in the form of Euler’s equation of motion $\boldsymbol{\tau} = \mathbf{M}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{M}\boldsymbol{\omega}$. This aspect of the dynamics [Rimon and Burdick, 2019] is particularly important for determining the controllability of a grasped object [Brook et al., 1998]. The inertia tensor is easily obtained from the scatter matrix (21) of the object [Selig and Martins, 2014]:

$$\mathbf{M} = \text{tr}(\mathbf{Q}) \mathbf{I}_3 - \mathbf{Q}. \quad (22)$$

This shows that if the rotation axis is parallel to a principal axis of the covariance ellipsoid, then the momentum vector \mathbf{m} is parallel to the angular velocity vector $\boldsymbol{\omega}$, because \mathbf{M} and \mathbf{Q} have the same eigenvectors (22). Note that the matrix $\mathbf{Q}(\mathcal{P})$, and the corresponding inertia tensor $\mathbf{M}(\mathcal{P})$ in (22), are expressed with respect to the overall centroid $\bar{\mathbf{p}}$. It may be necessary to express the inertia tensor in relation to a different point, which can be done via the parallel axis theorem [Arnold, 1989].

3.4 Discrete curvature

It will be useful to measure the surface curvature of a polycube object, for use in the grasp generation procedure. Rather than discretizing the classical definition, it is more direct to consider a discrete analogue, defined over the creases of the polyhedral object. Suppose that triangles \mathcal{T}_i and \mathcal{T}_j share the vertex indices $[\ell_i^a, \ell_i^b]$, taken in anticlockwise order from the three vertex indices $[\ell_i^1, \ell_i^2, \ell_i^3]$ of \mathcal{T}_i . Then the dihedral angle θ_{ij} in the range $\pm\pi$ can be obtained [Crane, 2023] by applying the four-quadrant arctan function to the formula

$$\tan \theta_{ij} = \frac{\mathbf{e}_{ij} \cdot (\mathbf{n}_i \times \mathbf{n}_j)}{|\mathbf{e}_{ij}| \mathbf{n}_i \cdot \mathbf{n}_j}, \quad \mathbf{e}_{ij} = \mathbf{x}[\ell_i^b] - \mathbf{x}[\ell_i^a]. \quad (23)$$

Convex and concave creases have positive and negative angles respectively, using these conventions. Note that the antisymmetry $\mathbf{e}_{ij} = -\mathbf{e}_{ji}$, which arises from the vertex ordering in adjacent faces, is cancelled by the corresponding antisymmetry of the cross product in (23).

\mathcal{P}	$ \mathcal{P} $	$A(\mathcal{P})$	$C(\mathcal{P})$	$D(\mathcal{P}_1, \mathcal{P})$
A	1	6	1.0	0.0
B	2	10	0.864	2.299
C	3	14	0.708	3.511
D	3	14	0.708	3.644
E	4	16	0.844	4.159
F	4	18	0.593	4.377
G	4	18	0.593	4.390
H	4	18	0.593	4.427
I	4	18	0.593	4.428
J	4	18	0.593	4.564
K	4	18	0.593	4.598

Table 2: Statistics of the test objects shown in the top row of figure 1, including volume $|\cdot|$, surface area A , compactness C , and shape distance D from unit cube \mathcal{P}_1 . The table is ordered by shape distance D . Note that the compactness C is not very discriminative.

Computationally, it is convenient to define a set of indices for the geometrically distinct object edges of the object:

$$\mathcal{E}(\mathcal{P}) = \{[i, j] : \mathcal{T}_i \sim \mathcal{T}_j, i < j\} \quad (24)$$

where ‘~’ denotes adjacency, and the indices are ordered $i < j$ to prevent double-counting of edges. The discrete mean curvature of an edge can now be defined as the dihedral angle itself, weighted by the edge length: $\bar{\kappa}_{ij} = \frac{1}{2} |\mathbf{e}_{ij}| \theta_{ij}$. The factor of $\frac{1}{2}$ arises because each angle is notionally averaged with the curvature *along* the edge, which is zero. Summing this measure over the entire surface gives the total discrete mean curvature

$$\bar{\kappa}(\mathcal{P}) = \frac{1}{2} \sum_{[i,j] \in \mathcal{E}} |\mathbf{e}_{ij}| \theta_{ij}. \quad (25)$$

Note that the curvature is sensitive to both the original arrangement of the cubes, and to the random perturbations of the faces.

3.5 Shape distance

A useful shape metric can be defined, based on the Riemannian distance [Förstner and Moonen, 2003] between the corresponding covariance matrices

$$D^2(\mathcal{O}, \mathcal{P}) := \sum_{k=1}^3 \log(\lambda_k(Q_{\mathcal{P}} Q_{\mathcal{O}}^{-1}))^2 \quad (26)$$

where $\lambda_k(\cdot)$ are the eigenvalues of the indicated matrix. This distance is based on the principal axes of the two shapes, which are effectively compared by taking their three symmetrized ratios. In particular, $D(\mathcal{P}_1, \mathcal{P})$ is the

distance from the unit cube \mathcal{P}_1 to the polycube \mathcal{P} . This metric is intuitively sensitive to the size and shape of \mathcal{P} , as indicated in figure 1 and table 2. Furthermore (26) is closely related to both the sampling scheme (11) and the object kinematics (22).

4 Contact synthesis

In order to generate more feasible grasps, it is desirable to sample the surface more densely in more promising parts of the object. In particular, if a sample point is located at a roof-like crease of a polyhedral object, then it is unlikely to afford a good contact for grasping (certainly in the case of hard contacts). The same may be true along extreme concave creases; e.g. although a 90° concave crease may seem attractive, it is typically better to make *perpendicular* contact on one of faces, adjacent to the crease (rather than approaching from 45° , and making two poor contacts with the object). An increasing function of the discrete curvature can be used to penalize the sampling process, for example

$$C(\theta) := \begin{cases} \sin^2(\theta) & \text{if } |\theta| < \frac{\pi}{2} \\ 1 & \text{otherwise.} \end{cases} \quad (27)$$

The effect of this penalty should be localized to a strip of width ϵ , on either side of the crease, where ϵ is related to the finger-width of the gripper. This can be modelled by a decreasing weight-function $B(t) \in [0, 1]$, where $t = d/\epsilon$ is the normalized distance of point x to the closest edge. In the absence of other considerations, the skew-symmetry $B(t) = 1 - B(1-t)$ is imposed on the shape of this boundary function. In particular, consider the form $b(t)/(b(t) + b(1-t))$, and set $b(t) = 1/t^\eta$, to obtain:

$$B_\eta(d, \epsilon) := \begin{cases} \frac{1}{1 + (\frac{t}{1-t})^\eta} & \text{if } t < 1, \text{ where } t = d/\epsilon \\ 0 & \text{otherwise.} \end{cases} \quad (30)$$

Note that $B(0) = 1$ on any face boundary, while the parameter η controls the steepness of the downward transition; in particular $\eta = 1$ gives a linear ramp, while $\eta \rightarrow \infty$ leads to a step-function at $d = \frac{\epsilon}{2}$. Related constructions can be found in the literature on bump functions [Tu, 2008].

The complete penalty is the product of the boundary and curvature functions, so that a point x is unlikely to be near any high-curvature edge. If the k -th edge is $\mathbf{e}_k(s)$, where s is the parameter along the segment, then the desired multiplier $R \in [0, 1]$ of the sampling intensity is

$$R(x, \epsilon) := 1 - \max_k \max_s B(|x - \mathbf{e}_k(s)|, \epsilon) C(\theta_k) \quad (31)$$

where k indexes the edges in \mathcal{E} from (24). The maximization over s is performed analytically, by finding the closest point $\mathbf{e}_k(s)$ to x , while the maximization over k

is done by examining all edges. Note that $R(\mathbf{x}, \epsilon) < 1$ if $B > 0$ (\mathbf{x} is near e_k) and $C > 0$ (e_k is creased). Efficiency can be improved by limiting these tests to the edges of the current and adjacent triangles, thereby including the important case in which a creased edge meets the current triangle at a single vertex.

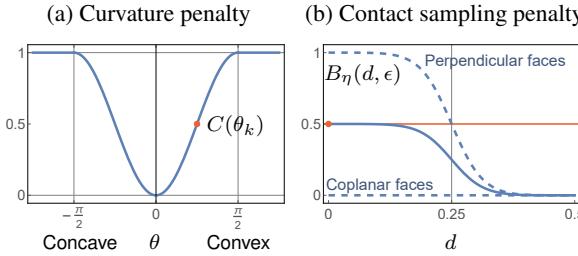


Figure 7: The curvature penalty (27) shown in (7a) is incurred at any crease in the mesh, as a function of the dihedral angle θ . The crease penalty (31) shown in (7b) reduces the contact sampling density as a function of proximity d to a mesh edge. It is defined by a soft step function (30), multiplied by the corresponding curvature penalty.

Rejection sampling can now be used to thin the existing points, obtained in section 2.4, so that their density is proportional to $R(\mathbf{x}, \epsilon) \in [0, 1]$, the complement of the penalty (31). The initial samples have *constant* density on the surface, so it suffices to reject a given point \mathbf{x} if $u > R(\mathbf{x}, \epsilon)$, where $u \sim \mathcal{U}_{[0,1]}$ is a uniform random variate. It follows that all samples in the interior (further than ϵ from an edge) or flat regions are retained, while the density falls off near any creased edge, at a rate controlled by the parameter η .

Note that it is straightforward to make the curvature penalty asymmetric, with respect to convex vs. concave edges. Two margin penalties ϵ^+ and ϵ^- are set for these cases, respectively. Then either $B(d, \epsilon^+)$ or $B(d, \epsilon^-)$ is evaluated, according to the sign of the angle θ_{ij} . For example, if ϵ^- is set to zero, then no points will be removed around concave creases.

5 Discussion

This paper has introduced a family of stochastic polycube objects, including an algorithm for dataset generation,

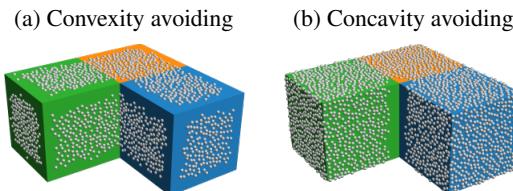


Figure 8: Curvature-sensitive sampling of grasp contact-points. Samples can be biased to avoid convex (8a) or concave (8b) parts of the shape.

and an analysis of the resulting shape properties. Future work will explore the intuitive idea of ‘graspability’, in relation to different 3D shapes, using the polycube models described here.

The role of surface curvature is another interesting topic to explore, in relation to robotic grasping. Although the shapes discussed here are not curved in the classical sense, they do exhibit complex discrete curvatures of all kinds. For example, consider the nine vertices of a 2×2 planar mesh face. The vertices can be perturbed to yield all types of Gauss curvature; peak/hollow, roof/valley, and saddle. It can be hypothesized that this variability is sufficient for effective transfer learning, in practice, despite the lack of continuous curvature variation.

More generally, it may be possible to analyze contemporary learning-based approaches to robotic grasping, by making use of mathematically tractable shape datasets, as proposed here.

References

- G. Aleksandrowicz and G. Barequet. Counting d -dimensional polycubes and nonrectangular planar polyominoes. In *Proc. COCOON*, pages 418–427, 2006.
- M. Andries, Y. Fleytoux, S. Ivaldi, and J.-B. Mouret. AGOD-Grasp: an Automatically Generated Object Dataset for Benchmarking and Training Robotic Grasping Algorithms. working paper or preprint, Mar. 2025. URL <https://inria.hal.science/hal-03983079>.
- V. Arnold. *Mathematical Methods of Classical Mechanics*. Springer, 1989.
- M. Berger. *Geometry Revealed: A Jacob’s Ladder to Modern Higher Geometry*. Springer, 2010.
- E. Berlekamp, J. Conway, and R. Guy. *Winning ways for your mathematical plays*, volume 4, chapter 24, pages 843–925. A.K. Peters, 2nd edition, 2004.
- K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *Proc. ICRA*, pages 4243–4250, 2018.
- R. Bridson. Fast Poisson disk sampling in arbitrary dimensions. In *Proc. SIGGRAPH*, page 22, 2007.
- N. Brook, M. Shoham, and P. Dayan. Controllability of grasps and manipulations in multi-fingered hands. *IEEE Trans. Robotics and Automation*, 14(1):185–192, 1998.

- B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar. Yale-cmu-berkeley dataset for robotic manipulation research. *International Journal of Robotics Research*, 36(3):261–268, 2017.
- M. Corsini, P. Cignoni, and R. Scopigno. Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE Trans. Visualization and Computer Graphics*, 18(6):914–924, 2012.
- K. Crane. Discrete differential geometry: an applied introduction, 2023.
- H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Trans. Graphics*, 13(1):43–72, 1994.
- M. Eden. A two-dimensional growth process. In *Proc. Fourth Berkeley Symp. on mathematical statistics and probability*, pages 223–239, 1961.
- S. Eslami, D. Rezende, F. Besse, F. Viola, A. Morcos, M. Garnelo, A. Ruderman, A. Rusu, I. Danihelka, K. Gregor, D. Reichert, L. Buesing, T. Weber, O. Vinyals, D. Rosenbaum, N. Rabinowitz, H. King, C. Hillier, M. Botvinick, D. Wierstra, K. Kavukcuoglu, and D. Hassabis. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.
- H.-S. Fang, C. Wang, M. Gou, and C. Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proc. CVPR*, pages 11441–11450, 2020.
- W. Förstner and B. Moonen. A metric for covariance matrices. In E. W. Grafarend, F. W. Krumm, and V. S. Schwarze, editors, *Geodesy: The Challenge of the 3rd Millennium*, pages 299–309. Springer, 2003.
- M. Gardner. Mathematical games. *Scientific American*, 227(3):176–182, 1972.
- M. Gardner. *Knotted doughnuts and other mathematical entertainments*. W.H. Freeman, 1986.
- A. Grundmann and H. M. Möller. Invariant integration formulas for the n -simplex by combinatorial methods. *SIAM Journal on Numerical Analysis*, 15(2):282–290, 1978.
- F. Harary. Graphical enumeration problems. In F. Harary, editor, *Graph Theory and Theoretical Physics*, chapter 1, pages 33–38. Academic Press, 1967.
- W. S. Harwin and A. Barrow. Multi-finger grasps in a dynamic environment. In *Multi-finger Haptic Interaction*, pages 5–30. Springer, 2013.
- J. Huang, T. Jiang, Z. Shi, Y. Tong, H. Bao, and M. Desbrun. L1-based construction of polycube maps from complex shapes. *ACM Trans. Graphics*, 33(3):1–11, 2014.
- K. Huebner, S. Ruthotto, and D. Kragic. Minimum volume bounding box decomposition for shape approximation in robot grasping. In *Proc. ICRA*, pages 1628–1633, 2008.
- Y. Jiang, S. Mosenon, and A. Saxena. Efficient grasping from RGBD images: Learning using a new rectangle representation. In *Proc. ICRA*, pages 3304–3311, 2011.
- L. Klein, G. Maiello, V. Paulun, and R. Fleming. Predicting precision grip grasp locations on three-dimensional objects. *PLoS Computational Biology*, 16(8), 2020.
- V. Lippiello, B. Siciliano, and L. Villani. Multi-fingered grasp synthesis based on the object dynamic properties. *Robotics and Autonomous Systems*, 61(6):626–636, 2013.
- W. Lunnon. Symmetry of cubical and general polyominoes. In R. C. Read, editor, *Graph Theory and Computing*, pages 101–108. Academic Press, 1972.
- H. Ma, M. Shi, B. Gao, and D. Huang. Generalizing 6-DoF grasp detection via domain prior knowledge. In *Proc. CVPR*, pages 18102–18111, 2024.
- J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. Ojea, and K. Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *Proc. RSS*, 2017.
- F. Manin, E. Roldan, and B. Schweinhart. Topology and local geometry of the Eden model, 2020. URL <https://arxiv.org/abs/2005.12349>.
- N. Mavrakis and R. Stolkin. Estimation and exploitation of objects’ inertial parameters in robotic grasping and manipulation: A survey. *Robotics and Autonomous Systems*, 124:103374, 2020.
- D. Morrison, P. Corke, and J. Leitner. EGAD! An Evolved Grasping Analysis Dataset for diversity and reproducibility in robotic manipulation. *IEEE Robotics and Automation Letters*, 5(3):4368–4375, 2020.
- R. Newbury, M. Gu, L. Chumbley, A. Mousavian, C. Eppner, J. Leitner, J. Bohg, A. Morales, T. Asfour, D. Kragic, D. Fox, and A. Cosgun. Deep learning approaches to grasp synthesis: A review. *IEEE Trans. Robotics*, 39(5):3994–4015, 2023.
- P. Ni, W. Zhang, X. Zhu, and Q. Cao. PointNet++ Grasping: Learning an end-to-end spatial grasp generation algorithm from sparse point clouds. In *Proc. ICRA*, pages 3619–3625, 2020.
- OEIS Foundation Inc. Number of 3-dimensional polyominoes (or polycubes) with n cells, in *The Online Encyclopedia of Integer Sequences*, 2025a. URL <https://oeis.org/A000162>.

OEIS Foundation Inc. Number of n -celled solid polyominoes (or free polycubes, allowing mirror-image identification), in The Online Encyclopedia of Integer Sequences, 2025b. URL <https://oeis.org/A038119>.

F. T. Pokorny, K. Hang, and D. Kragic. Grasp moduli spaces. In *Proc. Robotics: Science and Systems*, page 36, 2013.

J. Ponce, S. Sullivan, A. Sudsang, J.-D. Boissonnat, and J.-P. Merlet. On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *The International Journal of Robotics Research*, 16(1):11–35, 1997.

E. Rimon and J. Burdick. *The Mechanics of Robot Grasping*. Cambridge University Press, 2019.

J. Selig and D. Martins. On the line geometry of rigid-body inertia. *Acta Mechanica*, 225:3073–3101, 11 2014. doi: 10.1007/s00707-014-1103-7.

R. Shepard and J. Metzler. Mental rotation of three-dimensional objects. *Science*, 171:701–703, 1971.

J. Tobin, L. Biewald, R. Duan, M. Andrychowicz, A. Handa, V. Kumar, B. McGrew, A. Ray, J. Schneider, P. Welinder, W. Zaremba, and P. Abbeel. Domain randomization and generative models for robotic grasping. In *Proc. IROS*, pages 3482–3489, 2018.

L. Tu. *An Introduction to Manifolds*. Springer, 2008.

G. Turk. Generating random points in triangles. In *Graphics Gems*, chapter I.5, pages 24–28. Academic Press, 1990.

G. Vezzani, U. Pattacini, and L. Natale. A grasping approach based on superquadric models. In *Proc. ICRA*, pages 1579–1586, 2017.

D. Wang, D. Tseng, P. Li, Y. Jiang, M. Guo, M. Danielczuk, J. Mahler, J. Ichnowski, and K. Goldberg. Adversarial grasp objects. In *Proc. IEEE CASE*, pages 241–248, 2019.