AGOD-Grasp: an Automatically Generated Object Dataset for benchmarking and training robotic grasping algorithms

Mihai Andries, Yoann Fleytoux, J.-B. Mouret, Serena Ivaldi

AGOD-Grasp: an Automatically Generated Object Dataset for benchmarking and training robotic grasping algorithms

Mihai Andrieș and Yoann Fleytoux and Serena Ivaldi and Jean-Baptiste Mouret

***Abstract*— Robust robotic grasping of objects has broad industrial applications. The reliability of data-driven grasping methods is influenced by the variability of object shapes encountered during training. Most existing objects datasets suffer from human selection bias, lack variability, or are non-reproducible. This paper presents a physically reproducible 3D-printable object dataset for training and evaluating grasping algorithms. It contains exact 3D meshes of 50 objects for simulation and printing purposes. The various objects in the dataset were found using the MAP-Elites algorithm, optimising the variability of objects according to two grasping metrics. We used a Variational AutoEncoder (VAE) as a generative model for voxelgrid object models, which were then converted to meshes and simplified using Volumetric Hierarchical Approximate Convex Decomposition (V-HACD). The dataset is publicly available online, and can be ordered from any 3d-printing service according to given specifications. We hope it will become a standard benchmarking dataset for the robotic grasping community.**

I. Introduction and Motivation

Grasping and manipulation are essential abilities for a robot. To compute grasps on objects, two broad approaches exist: (1) analytical (model-based) grasping, which requires information about the manipulated object (weight, center of mass, friction) that may not be readily available, and (2) data-driven approaches, which exploit pre-recorded or simulated grasping data to sample potential grasps and rank them. The efficiency of data-driven approaches depends, among others, on the diversity of objects seen during training. Most object datasets for training and evaluating robotic grasping consist of objects made by humans for humans. This negatively impacts the grasping of broken or unusual objects, or trash in general, which is often the use-case for robots in the recycling industry. Moreover, standardized object datasets require logistic efforts to provide and acquire them, and depend on availability of these objects on the market.

This paper presents a 3d-printable object dataset designed for evaluating and training robotic grasping algorithms. The dataset was conceived so as to contain objects of various grasping complexity, according to two grasping metrics employed. We provide the mesh models (in stereolithography (STL) format) for 3d-printing the objects to make the benchmark reproducible, and for tests in simulation. Laboratories lacking 3D-printing capabilities can order the object dataset from a 3D printing service.

This dataset can be used in scenarii where objects lack a definite prior-known shape, like waste sorting and recycling, management of contaminated materials from nuclear power plants, as well as in more traditional use-cases like domestic cleaning robots, bin picking for warehouse robots, etc.

Université de Lorraine, CNRS, Inria, LORIA, Nancy, France
mihai.andries@inria.fr

The remainder of the paper is organised as follows: Section II presents the related work on object datasets for grasping, Section III describes the methodology employed for generating the object dataset, Section IV displays our results with an image of the dataset and with plots describing the shape exploration progress, and we conclude in Section V with opportunities for future work.

II. Related work

In this section, we describe the desired properties for an object dataset, and then analyse existing datasets according to these properties.

A. Desired dataset properties

We are interested in the following properties for an object manipulation datasets:

- object form variability (to counter human bias in object design);
- availability of 3D models for simulation purposes (with complete object models preferred over 3D scans);
- availability of physical versions of objects (in addition to digital 3D models);
- reproducibility, defined by either (a) the possibility to make exact copies, usually using printable 3D models; or (b) distribute-ability: keeping large object stocks provided by reliable and long-lived suppliers, and the ability to deliver an object set (if it cannot be 3d-printed).

B. Existing object model datasets for grasping evaluation

This section lists object datasets suitable for training and evaluating grasping algorithms. Table I contains a brief description of each dataset in chronological order of their publication, while Table II analyses them according to our criteria of interest mentioned above in Section II-A. A detailed review of existing object model datasets was provided by Calli et al. [4]. Datasets specifically designed for grasping were analysed by Morrison et al. in [19].

Choi et al. created a list of objects that are difficult to manipulate by motor-impaired patients [5]. The list illustrates categories of objects presenting manipulation difficulties, but does not contain numerical representations of these objects. Similarly, the KIT object models database [11] contains object models obtained from 3D scans, but is not physically reproducible. Jiang et al. [10] provided a grasping dataset containing RGB-D images of objects, each acccompanied by

a few positive and negative grasping rectangles. However, no physical objects are provided, nor models for 3D printing.

The Amazon picking challenge [7] involved manipulation of objects commonly sold online. The report describes the practical challenges encountered by robots due to objects of deformable shape, transparent or reflective surfaces, wire-frame objects, etc. The list is valuable due to its practical insights on what makes an object difficult to grasp, but does not constitute an object set.

RoboCup@Home is another competition for domestic robots involving manipulation of objects (both known and unknown). The list of predefined objects includes kitchen-ware, bags, trays, deformable and fragile objects, etc. The object set is of practical interest, but is not exactly reproducible for a benchmark setting outside of the competition.

Google presented a dataset of procedurally generated random rigid objects [21], provided in OBJ and URDF representations for use with the Bullet physics simulator. Since objects were generated according to a single formula, the dataset lacks diversity.

The YCB object and model set [1] has become widely used in the robotics community. However, obtaining a copy takes up to several months due to logistics, and encounters cross-border shipping difficulties (containers with food items, like cans and biscuit boxes, are emptied before shipping) and supply difficulties (some original objects were replaced with similar ones). The object meshes were obtained from object scans, and are not exact numerical representations (this applies in particular to articulated and flexible objects like chains and ropes).

The work most related to ours is that of Morrison et al. [19], who generated a dataset of 2000 objects for evaluating and training grasping algorithms, using Compositional Pattern Producing Networks (CPPNs) which produce objects with radial symmetry. Of these, the authors subsequently selected and post-processed a set of 49 objects that are printable without support structures using Fused filament fabrication (FFF) 3d-printing. Objects are evaluated using the Ferrari-Canny metric provided by DexNet 2.0 [15]. Similarly, the authors used the MAP-Elites algorithm [20] to guarantee variability in the generated dataset.

Our work differs in two important aspects:

- for the generative model, we use a VAE with 3D convolutions, which processes voxelgrid models of size $64 \times 64 \times 64$,
- for evaluating the generated objects, we use two metrics: (1) one based on the distance between the grasp centroid and the object centroid, and (2) one based on the angular distance between the grasp axis and the face normals passing through the contact points.

III. Methodology

This section presents the methodology employed for generating the object set.

A. Pipeline

The object shape generation pipeline, shown in Figure 2, has several components:

1) the sampling algorithm (MAP-Elites [20]), which generates a value vector encoding an object shape;
2) a generative model, which transforms a value vector into a 3D voxelgrid shape;
3) the voxelgrid-to-mesh converter, which selects the biggest connected component inside the generated voxelgrid, and converts it into a mesh using the Marching Cubes algorithm [14];
4) the mesh simplifier (V-HACD library [17] integrated into PyBullet), which simplifies the mesh by approximating it with a set of convex parts;
5) the mesh graspability evaluator, which provides two graspability scores for a two-fingered parallel gripper:
   - a score based on the distance between the grasp centroid and the mesh centroid, and
   - a score based on the angular distance between the grasp axis and the normals of the faces on which the grasp points are located.

B. Sampling algorithm

We use the MAP-Elites algorithm [20] to sample the search space of object shapes, defined by the latent layer of a VAE described in the next section. MAP-Elites generates value vectors encoding object shapes, feeds these vectors into the pipeline that converts them into 3d object shapes and evaluates their graspability scores, and then projects these objects into a 2D space defined by two grasp-ability metrics. We subdivide the projection space, defined by two object grasp-ability metrics, into 2^{14} niches. Dividing the space into many regions allows to have more niches for which samples compete, thus obtaining a more diverse population.

MAP-Elites being an evolutionary exploration algorithm, it needs an initial population from which to start its exploration. The goal of the MAP-Elites initialisation phase is to populate with samples 100 niches, in order to have some diversity to start the evolutionary exploration. We seed the initial population with (1) average latent representations of objects from all the categories of the ModelNet 40 dataset, (2) their pairwise combinations, and (3) 10 latent object representations sampled uniformly from the $[-2,2]$ interval. Since the KL divergence loss of the VAE forces the latent representations to resemble a $(\mu = 0, \sigma = 1)$ gaussian, 95% of samples should fall within this interval (which corresponds to $\mu \pm 2\sigma$) according to the 68–95–99.7 rule. MAP-Elites then continues to sample shapes from the gaussian distribution $(\mu = 0, \sigma = 1)$ until it either fills 100 niches with samples, or until it reaches 4000 sampled objects. After this phase, MAP-Elites continues with its evolutionary search phase, involving random and directed mutations.

C. Shape generation

As a generative model, we use a VAE [12, 22] with an Encoder composed of 3d-convolutional layers, and a Decoder composed of transpose 3d-convolutions. The network layers

TABLE I: Object datasets suitable for grasping purposes

Dataset	Description	Format	Year
Matamoros et al. [18]	Tableware, cutlery, trash bags, bags, disks, books, coat rack, trays, pourable (muesli, cereal), heavy objects, tiny objects, fragile objects, amorphous objects	Physical objects, handed to Robocup participants	2009-2019
Choi et al. [5]	Medical items, dining items, bathroom items, living room items, bed room items, personal belongings	Shopping list (outdated)	2009
Jiang et al. [10]	1035 images of 280 different objects (different orientations, associated point cloud, background image)	The raw dataset consists of: (a) images, (b) grasping rectangles, (c) pointclouds, (d) background images and (e) a file containing a mapping from each image to the corresponding background image.	2011
Kasper et al. [11]	Boxes, cans, tubes, mugs, bottles, toys, pliers. Limited variation in object types. 3D models are reconstructions, not sources.	3D triangulated mesh, 2D image data, texture information, grasp information.	2012
Calli et al. [1, 3, 2]	Food items, kitchen items, tools, shape imate,s task items (toys, magazine)	Point clouds, associated meshes and texture, mass, dimensions, models for integration into planning and simulation software. Meshes reconstructed from scanning (without bottom side of objects).	2015
Correll et al. [7]	25 products commonly sold on Amazon, with various shapes, sizes, deformable shapes, transparent and reflective surfaces, fragile.	Shopping list of 25 objects commonly sold on Amazon.	2016
Google [21]	Procedurally generated dataset	3D models of 1000 objects in Wavefront OBJ format.	
Morrison et al. [19]	Objects automatically generated using CPPNs	2000 objects in total, of which 49 objects are printable without supporting structures	2020
AGOD-Grasp dataset (proposed)	Objects automatically generated in voxelgrid format using a VAE, then converted to meshes.	50 objects selected out of several hundred thousand objects.	2020

TABLE II: Object datasets analysed by criteria of interest

		Object variability	RGB-D scans	Simulation models	Physical models	3D printable	Physically deliverable
[18]	Robocup @home	✓	✗	✗	✓	✗	✓
[5]	ALS dataset	✓	✗	✗	✗	✗	✗
[10]	Cornell grasping dataset	✓	✓	✗	✗	✗	✗
[11]	KIT object set	✗	✓	✓	✗	✗	✗
[1]	YCB dataset	✓	✓	✓	✓	✗	✓
[7]	Amazon picking challenge	✓	✗	✗	✓	✗	✓
[21]	Google procedurally generated dataset	✗	✗	✓	✗	✓	✗
[19]	EGAD dataset	✓	✗	✓	✓	✓	✗
	AGOD-Grasp dataset	✓	✗	✓	✓	✓	✗

are interspersed with batch normalisation and rectified linear units (ReLUs). The VAE receives as input binary voxelgrid representations of objects with a resolution of $64 \times 64 \times 64$ pixels, and outputs binary voxelgrids of identical size. The dimension of the latent layer is 512, which is reshaped into a $8 \times 8 \times 8$ cube by the first layer of the decoder, and then upscaled through several layers to the final output size. Consequently, the search space has 512 dimensions, as it is the size of the value vector encoding an object shape. We used a weighted reconstruction loss, penalizing stronger the network for errors in reconstructing full voxels. This ensures that the reconstructions of voxelgrid object models (which generally contain more empty voxels that filled ones) are not completely empty at the start of training. The VAE was trained to reconstruct 3D shapes from all the object categories of the ModelNet40 object dataset [23]. This ensures that the network has sufficient generative power to create an (arguably) large enough variety of shapes.

The voxelgrid generated by the VAE is rescaled to fit within a cube with sides equal to the width of the gripper, then is converted into a mesh, by first selecting the biggest connected component in the voxelgrid, and then applying the Marching Cubes algorithm [14] implemented in the TriMesh python library [9]. Since meshes converted from voxelgrids tend to have rugged surfaces, we smooth them using the V-HACD algorithm [17] integrated into PyBullet [8], thus reducing the number of mesh faces while maintaining the overall shape. By tweaking the parameter constraining the minimum volume of convex components of the mesh, it allows to thicken object parts that are too thin and can break during 3d-printing or manipulation.

D. Grasp computation

Grasp candidates are generated using the normals of mesh faces (see Figure 1). 256 randomly selected face normals are used as grasp axes, and for each grasp axis we compute its points of intersection with the mesh, using the TriMesh [9] library, and select the two outermost points as grasp locations. We do this to avoid computing approach paths for grasping, to simplify the computation. If the distance between the two points is larger than the maximum gripper opening, this grasp candidate is discarded.

E. Grasp metrics

We use two metrics for evaluating grasps. The first metric, called *centroid distance* for brevity, computes the distance between the grasp centroid and the mesh centroid. The grasp centroid is defined as the center point between the grasp contact points. Since we use a parallel gripper with two

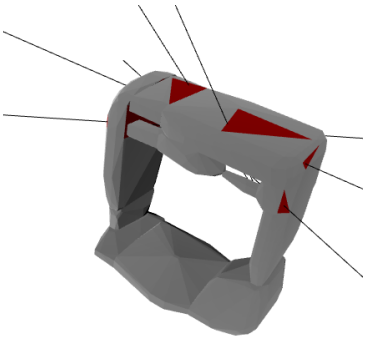


Fig. 1: Generating grasp candidates: grasp axes are generated from normals of mesh faces. The faces they pierce are coloured in red.

fingers and compute a single contact point for each finger, the grasp centroid is located at the middle of the segment linking the two contact points. We normalise this value, dividing it by the length of the bounding box diagonal (which is the maximum possible distance between the mesh centroid and a grasp centroid). This makes the metric invariant to object size.

The second metric, called *angular distance* for brevity, computes the angular distance between the grasp axis and the normals of the two faces on which the contact points are located. More precisely, we calculate the angle in radians between the grasp axis and each of the two face normals. We then normalise each angle, dividing it by $\Pi/2.0$, which corresponds to the largest possible angle between the face normal and an arbitrary line (coming out of the face plane) intersecting it. We sum the squares of these two angles, and normalise again, dividing by two.

F. Grasp evaluation

Each grasp is evaluated according to the two metrics mentioned above. For each grasp metric, the final graspability score of the object is computed as the 75th percentile value of the sampled grasps' scores.

MAP-Elites works on a normalised projection space, which in our case is a 2D space defined by two grasping metrics, contained within $[0..1] \times [0..1]$. It may happen that some regions of the projection space remain empty, as no objects are found that would fit in those regions. Since we compute the 75th percentile score for our grasping metrics, the samples cluster together more (in the projection space) than when using the best grasp score for each object. As samples compete for fewer regions of the search space, this reduces the population size, impeding the optimisation of the evolutionary algorithm. To counter this clustering effect, we spread the samples in the projection space, using the 4-th root of their grasp scores:

$$\text{spreaded grasp score} = \sqrt[4]{\text{grasp score}} \tag{1}$$

We also attempted to use grasp metrics computed by DexNet, but despite considerable effort, we could not replicate the set-up of DexNet 2.0 [15], which was the last version of DexNet capable of computing grasps on an object mesh, rather that on point clouds (as in later versions). Its online alternative, DexNet as a Service (DNaaS) [13], albeit functional for computing a single grasp metric, hanged when queried for two different grasp metrics on the same object mesh.

G. Object selection

After running MAP-Elites long enough for the exploration to reach a plateau (when the number of projection space regions filled with objects stop increasing), we cluster the selected objects using K-Means into 50 clusters. For each cluster, we save the object with the smallest euclidean distance to the cluster centroid. This forms the resulting object set.

IV. Results and discussion

The dataset-generation pipeline filled with objects 575 niches, of which we selected 50. Figure 3 shows the exploration progress, defined by the number of niches filled after each generation of the evolutionary algorithm.

Figure 4 shows the generated dataset. The bias due to the limited number of object categories in the training set (40 object categories, mostly furniture objects) is arguably visible in the resulting shapes with some objects seemingly being composed of parts of furniture objects.

The dataset is available online [1]. For physical experiments, it should be printed out of Polylactic acid (PLA) with a wall tickness of 1.2mm, and a 15% infill density with uniform inner distribution. This wall width was chosen so as to fit printing nozzles of diameters 0.4mm and 0.6mm. The code used for generating the object set is available online [2].

A. Comparison with the state-of-the-art

We compared our object set with EGAD [19], as it is the closest in purpose and methods employed. For a fair comparison, we rescaled the EGAD objects to the same maximum size that we used ($0.07m$ for each side of the object bounding box, corresponding to the maximum width the gripper), and compared it to our object set according to the grasp metrics described in Section III-E. Figure 6 shows a comparison between the niches filled by our dataset, and those filled by EGAD. Although our algorithm filled fewer niches, the generated objects are of interest, as they do not have radial symmetry compared to EGAD objects. Figure 5 illustrates some of the shapes generated during exploration.

B. Trials in simulation

We evaluated the set of generated objects in simulation, by attempting to grasp them using the GQCNN [16] grasping algorithm.

[1] https://github.com/heap-chist-era/object_grasping_dataset/

[2] https://gitlab.inria.fr/CHIST-ERA-Heap/shape-generation-code

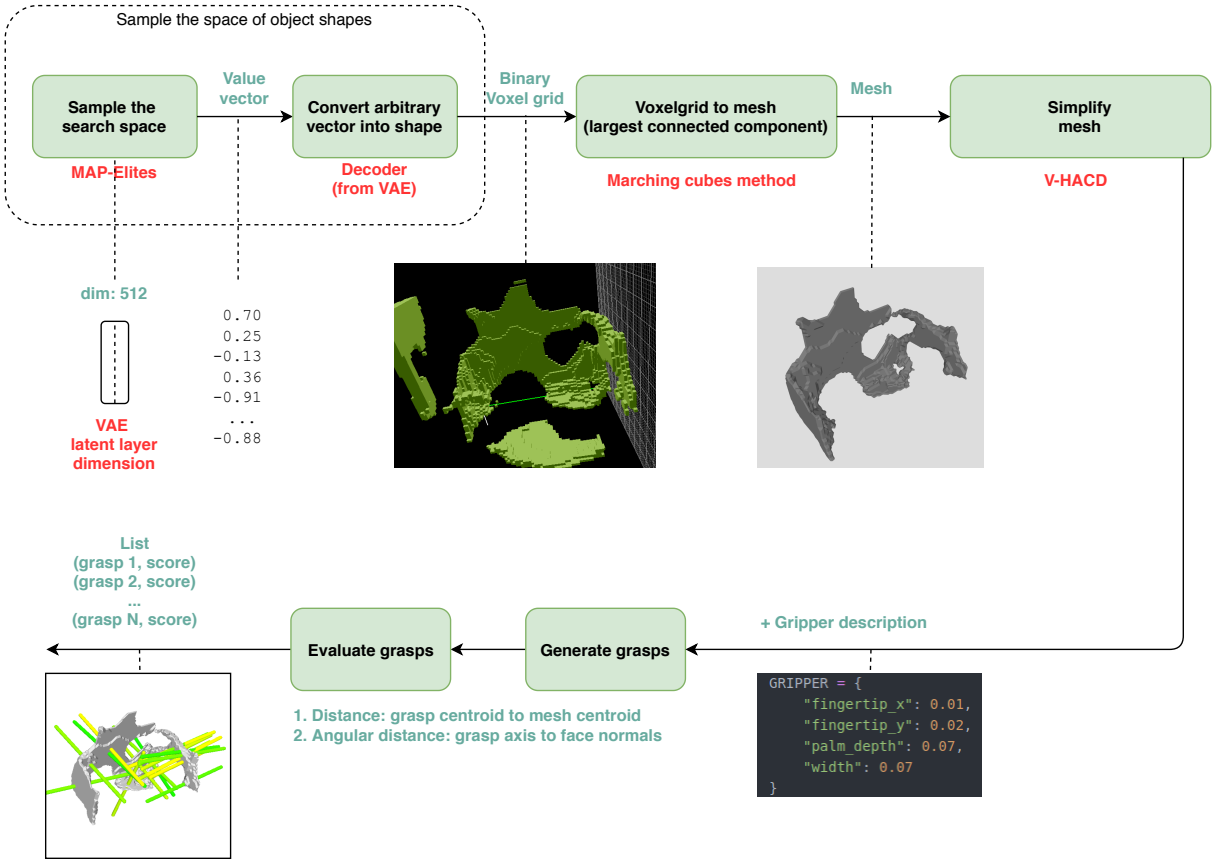


Fig. 2: The pipeline for object shape generation and graspability evaluation.

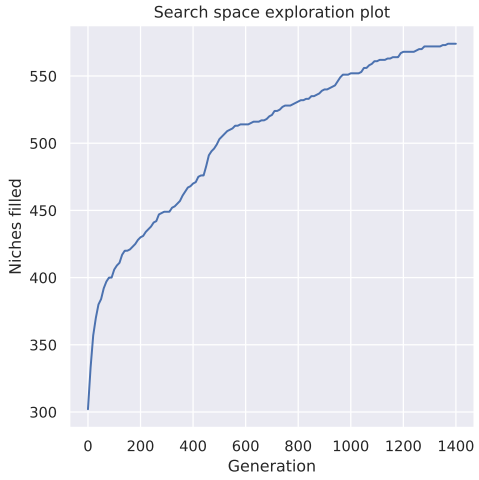


Fig. 3: Exploration progress, showing the number of niches filled against the number of generations. Generation 0 happens after the initialisation phase of the MAP-Elites algorithm, in which seed objects are randomly generated for the evolutionary exploration. Each generation explores 128 new object samples.

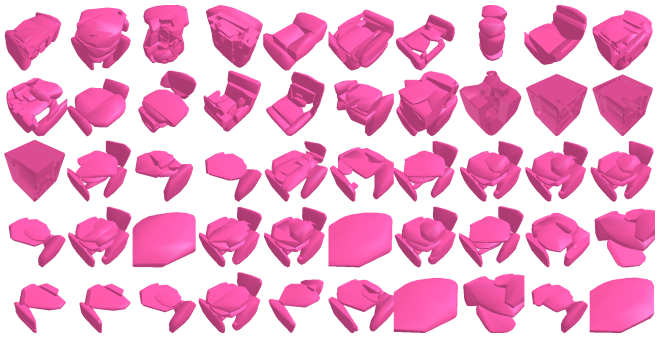


Fig. 4: 50 objects selected from the generated dataset using K-Means with 50 clusters. Objects were scaled for illustration purposes (i.e. their relative size is not correct).

V. Conclusion and Future Work

We generated a 3d-printable dataset of object models for training and evaluating robotic grasping algorithms. The dataset was optimised for the diversity of objects, according to two grasp metrics. The dataset is available online at `https://github.com/heap-chist-era/object_grasping_dataset/`.

In the future, we would like to investigate the graspability of articulated objects and objects composed of multiple materials, attributes which influence grasp stability. Future

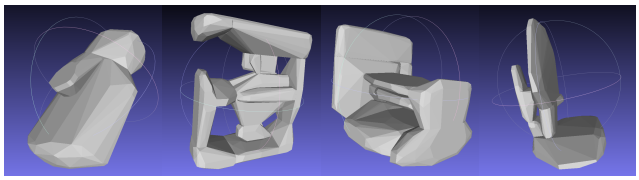
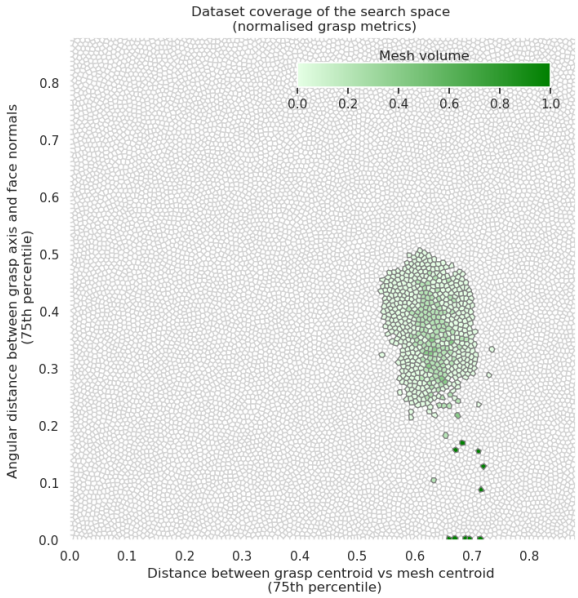
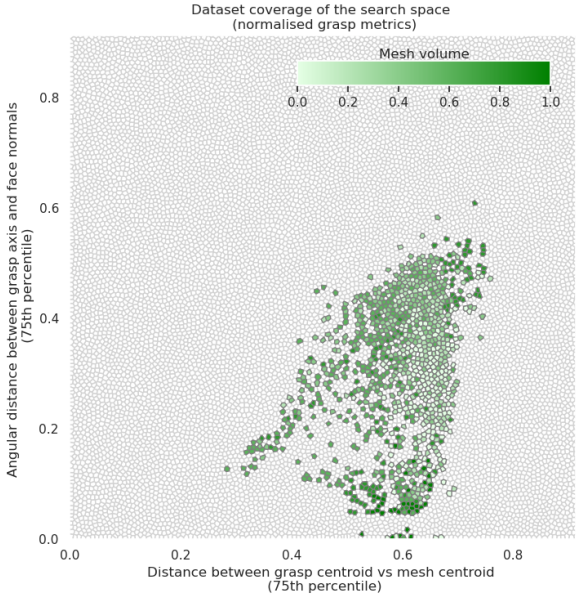


Fig. 5: Samples of generated objects. Viewer: MeshLab [6]



(a) Our dataset



(b) EGAD [19]

Fig. 6: Coverage of the projection space

work may also inspect the influence of:

- the shape representation employed, like shape programs and Constructive solid geometry (CSG),
- the diversity of objects in the training set of the generative model,
- the grasp sampling method.

VI. Acknowledgement

This work was made possible using funds provided by the EU H2020 ERA-NET funding scheme through the Chist-Era project HEAP.

References

[1] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar. "Benchmarking in Manipulation Research: Using the Yale-CMU-Berkeley Object and Model Set". *IEEE Robotics Automation Magazine* 22.3 (Sept. 2015), pp. 36–52. ISSN: 1070-9932. DOI: 10.1109/MRA.2015.2448951.

[2] Berk Calli, Arjun Singh, James Bruce, Aaron Walsman, Kurt Konolige, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. "Yale-CMU-Berkeley dataset for robotic manipulation research". *The International Journal of Robotics Research* 36.3 (2017), pp. 261–268. DOI: 10.1177/0278364917700714. eprint: https://doi.org/10.1177/0278364917700714. URL: https://doi.org/10.1177/0278364917700714.

[3] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. "The ycb object and model set: Towards common benchmarks for manipulation research". *2015 international conference on advanced robotics (ICAR)*. IEEE. 2015, pp. 510–517.

[4] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. "Benchmarking in manipulation research: The YCB object and model set and benchmarking protocols". *arXiv preprint arXiv:1502.03143* (2015).

[5] Young Sang Choi, Travis Deyle, Tiffany Chen, Jonathan D Glass, and Charles C Kemp. "A list of household objects for robotic retrieval prioritized by people with ALS". *2009 IEEE International Conference on Rehabilitation Robotics*. IEEE. 2009, pp. 510–517.

[6] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. "MeshLab: an Open-Source Mesh Processing Tool". *Eurographics Italian Chapter Conference*. Ed. by Vittorio Scarano, Rosario De Chiara, and Ugo Erra. The Eurographics Association, 2008. ISBN: 978-3-905673-68-5. DOI: 10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136.

[7] Nikolaus Correll, Kostas E Bekris, Dmitry Berenson, Oliver Brock, Albert Causo, Kris Hauser, Kei Okada, Alberto Rodriguez, Joseph M Romano, and Peter R Wurman. "Analysis and observations from the first amazon picking challenge". *IEEE Transactions on Automation Science and Engineering* 15.1 (2016), pp. 172–188.

[8] Erwin Coumans and Yunfei Bai. *PyBullet, a Python module for physics simulation for games, robotics and machine learning*. http://pybullet.org. 2016–2020.

[9] Dawson-Haggerty et al. *Software: Trimesh*. Version 3.2.0. 2019. URL: https://trimsh.org/.

[10] Yun Jiang, Stephen Moseson, and Ashutosh Saxena. "Efficient grasping from rgbd images: Learning using a new rectangle representation". *2011 IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 3304–3311.

[11] Alexander Kasper, Zhixing Xue, and Rüdiger Dillmann. "The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics". *The International Journal of Robotics Research* 31.8 (2012), pp. 927–934.

[12] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". *International Conference on Learning Representations*. 2014.

[13] Pusong Li, Bill DeRose, Jeffrey Mahler, Juan Aparicio Ojea, Ajay Kumar Tanwani, and Ken Goldberg. "Dex-Net as a Service (DNaaS): A Cloud-Based Robust Robot Grasp Planning System". *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE. 2018, pp. 1420–1427.

[14] William E Lorensen and Harvey E Cline. “Marching cubes: A high resolution 3D surface construction algorithm”. *ACM siggraph computer graphics*. Vol. 21. 4. ACM. 1987, pp. 163–169.

[15] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. “Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics” (2017).

[16] Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, and Ken Goldberg. “Learning ambidextrous robot grasping policies”. *Science Robotics* 4.26 (2019), eaau4984.

[17] Khaled Mamou. *Volumetric Hierarchical Approximate Convex Decomposition (V-HACD)*. Accessed: 2020-06-015. 2014. URL: https://github.com/kmammou/v-hacd.

[18] Mauricio Matamoros, Caleb Rascon, Sven Wachsmuth, Alexander William Moriarty, Johannes Kummert, Justin Hart, Sammy Pfeiffer, Matthijs van der Brugh, and Maxime St.-Pierre. *RoboCup@Home 2019: Rules and Regulations (draft)*. http://www.robocupathome.org/rules/2019_rulebook.pdf. 2019.

[19] Douglas Morrison, Peter Corke, and Jürgen Leitner. “EGAD! an Evolved Grasping Analysis Dataset for diversity and reproducibility in robotic manipulation”. *IEEE Robotics and Automation Letters* (2020).

[20] Jean-Baptiste Mouret and Jeff Clune. “Illuminating search spaces by mapping elites”. *arXiv preprint arXiv:1504.04909* (2015).

[21] *Procedurally generated random objects*. URL: https://sites.google.com/site/brainrobotdata/home/models (visited on 05/27/2019).

[22] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. “Stochastic Backpropagation and Approximate Inference in Deep Generative Models”. *International Conference on Machine Learning*. 2014, pp. 1278–1286.

[23] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. “3D ShapeNets: A deep representation for volumetric shapes”. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 1912–1920.