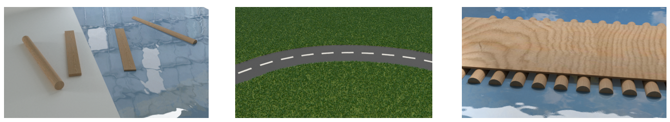
Generating object forms with desired affordances

Mihai Andries
Institute for Systems and Robotics
Instituto Superior Técnico
Lisbon, Portugal
mandries@isr.tecnico.ulisboa.pt

Atabak Dehban
Champalimaud Centre for the Unknown &
ISR, Instituto Superior Técnico
Lisbon, Portugal
adehban@isr.tecnico.ulisboa.pt

José Santos-Victor
Institute for Systems and Robotics
Instituto Superior Técnico
Lisbon, Portugal
jasv@isr.tecnico.ulisboa.pt



(a) *Wooden* beams have the *float-ability* affordance.

(b) *Flat* roads have the *traverse-ability* affordance.

(c) *Flat wooden* roads offer both *float-ability* and *traverse-ability*.

Fig. 1: The features that describe (a) wooden beams and (b) flat roads can be combined, to obtain an object design that possesses both *float-ability* and *traverse-ability*: (c) a pontoon bridge.

***Abstract*—Few Computer-Aided Design tools exist for exploring the design solution space. We introduce an algorithm for generating object forms with desired affordances. We follow the principle *form follows function*, and assume that object forms are related to affordances they provide (their functions). First, we use an artificial neural network to learn a function-to-form mapping from a dataset of affordance-labeled objects. Then, we combine forms providing desired affordances, generating object forms expected to provide all of them. We verify in simulation whether generated objects indeed possess the desired affordances by executing affordance tests. Examples are provided using the affordances contain-ability, sit-ability, and support-ability.**

I. Motivation

Traditionally, research in autonomous robots deals with the problem of recognising affordances of objects in the environment: i.e. given an object, what affordances does it offer? This paper addresses the inverse problem: given some affordances, what object form would provide them? (DARPA, 2017) This paper presents a method for automatic generation of object forms with desired affordances, which automatically relates object forms to their affordances, and then applies this knowledge to conceive new object forms that satisfy given functional requirements. Fig. 1 illustrates the concept of combining *features* describing two different objects to create another object possessing the *affordances* of both initial objects.

II. Related literature

A standard practice in design is to consult *knowledge ontologies* (Bryant et al., 2005) that contain function-to-form mappings (Umeda et al., 1997). Hu et al. (2018) presented a related review on object functionality inference from shape information. Autodesk, Inc. (2017) employed generative design to explore the space of 3D object shapes using genetic algorithms (Bentley, 1996).

To generate 3D forms from descriptions, modern techniques employ Auto-Encoders (Girdhar et al., 2016) and Generative Adversarial Networks (J. Wu et al., 2016), which learn a mapping from a low-dimensional probabilistic latent space to the space of 3D objects. Tian et al. (2019) proposed shape programs to represent 3D object models composed of multiple parts. In this study, we used a Variational AutoEncoder (VAE) (Kingma et al., 2014; Rezende et al., 2014) to both extract features describing 3D objects, and reconstruct the 3D shape of an object when given such a description.

A field of research that also focuses on linking object shapes with their affordances is that of *affordance learning*. It is based on the notion of *affordance* that defines an action that an object provides (or affords) to an agent (Gibson, 1977). In the context of this paper, we are interested in approaches that map object features to corresponding object affordances. Zech et al. (2017). published a review on affordances in cognitive robotics.

III. Methodology

The main idea is to train a VAE to reconstruct voxelgrid object models, and then generate novel shapes by combining latent codes from existing examples with desired affordances. The working hypotheses are: (i) objects providing the same affordance have common form features, (ii) averaging over multiple forms that provide the same affordance will extract a *functional form* providing that affordance, (iii) parametric interpolation between samples can generate novel forms with combined affordances of those samples. This last assumption is contentious, as we cannot yet predict the behaviour of affordances when combining their underlying shapes. Thus, we verify the presence of these affordances in simulation.

For simplicity, we employed a voxelgrid representation for 3D object models. The neural network is a Variational AutoEncoder with 3D Convolutional layers and a bottleneck latent layer, taking as input voxelgrid models of dimension 64x64x64, trained on ModelNet40 object dataset (Z. Wu et al., 2015) using a weighted reconstruction loss, penalising the network more strongly for errors in reconstructing full voxels.

Our workflow is composed of two phases: (1) learning phase, in which a neural network is trained to generate feature-based representations of objects and to faithfully reconstruct objects using this representation, and (2) request phase, in which a user

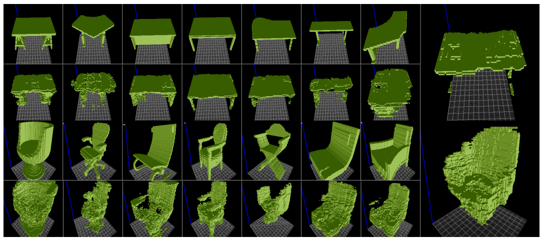


Fig. 2: Sample chairs/tables (top), their reconstructions (bottom), and the extracted *functional forms* (right). Visualiser: viewvox (Min, 2004).

requests the generation of a novel object with some desired affordances among those present in a menu. The algorithm then picks object categories providing those affordances, extracts corresponding shape features, and combines them to generate a feature description of a new object, which is then converted into a 3D voxelgrid model.

Every category of objects possesses a set of affordances that defines it. From a *form follows function* perspective, all object samples contained in a category share a set of features that provide its set of affordances. We call this set of features the *functional form* of a category of objects, computed as the average latent vector of an object category. We visualise it by inputting the obtained latent-vector description into the decoder trained to reconstruct 3D volumes (Fig. 2).

To combine two object descriptions, we compute how important is each variable in the description vector for encoding the object shape, by comparing it to corresponding variable describing a void volume and a non-informative prior distribution. We then combine these variables using a conflict resolution rule, giving priority to more important variables or averaging between them.

IV. Results and Discussion

In this section we provide results on the (a) capacity of the VAE to describe and reconstruct objects (Fig. 2), (b) extraction of functional forms for different object categories, (c) generation of novel objects through the combination of feature representations of object categories containing desired affordances, and (d) affordance testing for the generated objects. Fig. 2 shows the extraction of functional forms, relating features like flatness to *support-ability*, and seats with *sit-ability*. Fig. 3 shows the combination of *sit-ability* and *contain-ability* extracted from toilets and bathtubs, interpretable as bidets. Fig. 4 shows the combination of *support-ability* and *contain-ability* affordances. The comparison of generated objects with most similar samples from the dataset suggests that generated objects differ from training set samples (Fig. 4d). Tests executed in simulation verify that generated objects indeed provide the requested affordances (Fig. 5).

V. Conclusion

We presented a method for generating objects with desired affordances, by extracting a form-to-function mapping from a dataset of objects, combining these forms in a latent feature space learned by a neural network, and generating 3D object

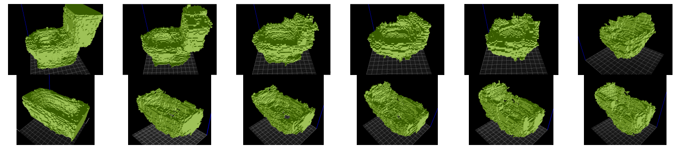
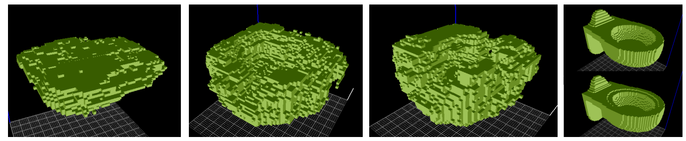


Fig. 3: Combination of functional forms with sit-ability and contain-ability, extracted from toilets and bathtubs, resembling a bidet.



(a) Flat functional form providing *support-ability*. (b) Concave form providing *contain-ability*. (c) Form providing *contain-ability* and *support-ability*. (d) Closest objects from training set.

Fig. 4: Combining features of objects providing respectively *contain-ability* and *support-ability*. (d) Closest objects from the training dataset.

models from them. We then test the presence of desired affordances in a physics simulator. Our models still lack information about materials from which objects are composed, and the articulations between subparts.

References

Autodesk, Inc. (2017). *Autodesk Dreamcatcher*. https://autodeskresearch.com/projects/dreamcatcher. Accessed: 2019-02-15.

Bentley, Peter John (1996). "Generic evolutionary design of solid objects using a genetic algorithm". PhD thesis. University of Huddersfield, UK.

Bryant, Cari R et al. (2005). "Concept generation from the functional basis of design". *ICED*. Engineers Australia, p. 1702.

DARPA (2017). *Disruption Opportunity Special Notice DARPA-SN-17-71*. https://www.fbo.gov/spg/ODA/DARPA/CMO/DARPA-SN-17-71/listing.html,

Gibson, James (1977). "The theory of affordances". *Perceiving, acting, and knowing: Toward an ecological psychology*, pp. 67–82.

Girdhar, Rohit et al. (2016). "Learning a Predictable and Generative Vector Representation for Objects". *Computer Vision – ECCV 2016*, pp. 484–499.

Hu, Ruizhen et al. (2018). "Functionality representations and applications for shape analysis". *Computer Graphics Forum*. Vol. 37. 2, pp. 603–624.

Kingma, Diederik P. et al. (2014). "Auto-Encoding Variational Bayes". *International Conference on Learning Representations*.

Min, Patrick (2004). *binvox*. `http://www.patrickmin.com/binvox`.

Rezende, Danilo Jimenez et al. (2014). "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". *ICML*, pp. 1278–1286.

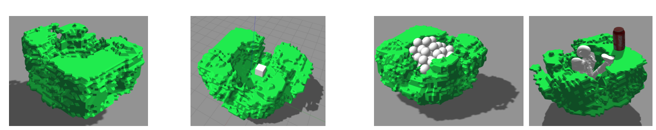
Tian, Yonglong et al. (2019). "Learning to Infer and Execute 3D Shape Programs". *International Conference on Learning Representations*.

Umeda, Yasushi et al. (1997). "Functional reasoning in design". *IEEE Expert*.

Wu, Jiajun et al. (2016). "Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling". *NIPS*, pp. 82–90.

Wu, Zhirong et al. (June 2015). "3D ShapeNets: A deep representation for volumetric shapes". *2015 IEEE CVPR*, pp. 1912–1920.

Zech, Philipp et al. (2017). "Computational models of affordance in robotics: a taxonomy and systematic classification". *Adaptive Behavior*.



(a) Generated object. (b) Supportability test. (c) Containability test. (d) Containability of a humanoid.

Fig. 5: Affordance tests for *support-ability* and *contain-ability*.