Conditional WGAN for grasp generation

Florian Patzelt, Robert Haschke and Helge Ritter
{fpatzelt,rhaschke,helge}@techfak.uni-bielefeld.de
Neuroinformatics Group, Center of Excellence Cognitive Interaction Technology (CITEC),
Bielefeld University, Germany*

Abstract. This work proposes a new approach to robotic grasping exploiting conditional Wasserstein generative adversarial networks (WGANs), which output promising grasp candidates from depth image inputs. In contrast to discriminative models, the WGAN approach enables deliberative navigation in the set of feasible grasps and thus allows a smooth integration with other motion planning tools. We find that the training autonomously partitioned the space of feasible grasps into several regions corresponding to different grasp *types*. Each region forms a smooth grasp manifold with latent parameters corresponding to important grasp parameters like approach direction.

We evaluate the model in simulation on the multi-fingered Shadow Robot hand, comparing it a) to a classical grasp planner for primitive geometric object shapes and b) to a state-of-the-art discriminative network model. The proposed generative model matches the grasp success rate of its trainer models and exhibits better generalization.

1 Introduction

Robotic grasping with artificial humanoid hands is a challenging and important research topic. Most recent approaches to robotic grasping involve a two-step procedure: First, a set of candidate grasps is sampled. This can either be done based on a previously extracted object representation [1, 2], heuristics [3] or just randomly around the object position [4–6]. In a second step, the most promising grasp is chosen based on a prediction of the expected grasp success.

In contrast to these approaches, the present work focuses on the rather unexplored area [7] of using generative models to *synthesize* grasps. In particular, we train a conditional generative adversarial network (GAN), which directly outputs promising grasp candidates from an input depth image in an end-to-end manner.

The proposed approach is faster than traditional discriminative approaches that need to sample and rank hundreds of grasps candidates. Furthermore, it provides a smooth grasp manifold suitable for deliberative search.

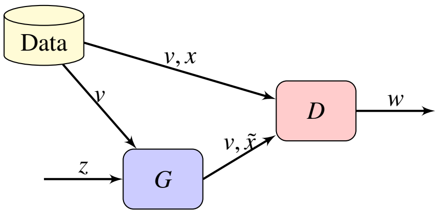

Fig. 1: Overview of the proposed WGAN architecture.

The proposed GAN comprises two subnetworks, the generator and the discriminator net, which are trained in an adversarial fashion (see Figure 1). Both sub-networks

*This work was supported by project FAMULA funded by the DFG Center of Excellence EXC 277: Cognitive Interaction Technology (CITEC).

receive the raw depth image v as input. While the discriminator additionally receives a grasp candidate x / $\tilde{x}$ as input, the generator takes a low dimensional latent parameter vector z to ensure a rich variability of grasps. The discriminator output w corresponds to the probability of the grasp candidate being from the training data distribution.

Training a GAN can be interpreted as learning a mapping from a latent space (comprising the noise vectors z) to the space of successful grasps, conditioned on the input depth image. As previous work on GANs suggest [8] a meaningful structure emerges for the learned mapping: The mapping is usually locally linear, points that are close in the latent space also map to points that are considered to be similar in the target space. Such a mapping in the grasp domain could be exploited to deliberatively search for promising grasps with different characteristics, e.g. power vs. precision grasps, or approaching the object from different directions.

The contribution of this paper is two-fold: First, we present a novel application of GANs to generate successful multi-fingered robotic grasps from depth images. Second, we provide an in-depth analysis of the learned mapping from latent-space to grasp space and highlight the emerged properties of this mapping.

2 Method

The goal of this work is to train a model that can sample successful grasps given a depth image input. The depth image $v_i \in V$ depicting object i is a 32×32 image. Grasps are represented by an 11-dimensional vector: 3 dimensions encode the position of the robot hand and 4 dimensions encode the orientation as a quaternion. The 4 remaining dimensions encode the desired grasp type using a one-hot vector. Namely, we consider Two-Finger-Precision, Three-Finger-Precision, All-Finger-Precision, and Power grasps. Each grasp type is associated with a fixed pre-grasp and grasp posture that determine how the fingers of the robot hand will close. Let X be the space of all grasps and $X_i \subset X$ the set of grasps that are successful on object i. Grasp success for this purpose is defined by whether the grasp is sufficient to lift the associated object in a MuJoCo [9] simulation.

Given these definitions, we can precisely formulate the goal of this work: We want to train a generative model that implements a mapping $G : Z \times V \to X$ such that $G(z, v_i) \in X_i \ \forall z \in Z \ \forall i$, i.e. that generates successful grasps for any object i given its depth image v_i and a uniform noise vector $z \in Z = [-1, 1]^n$. Furthermore, we want a model that can sample the full space of successful grasps i.e. $\forall x \in X_i$ there is a $z \in Z$ such that $x = G(z, v_i)$.

The GAN is trained in the typical adversarial manner. The generator G tries to produce grasps that the discriminator cannot distinguish from successful grasps given as training data thus approximating the distribution of successful grasps. The discriminator D works as adversary trying to distinguish between grasps from the generator and grasps from the set of successful grasps used as training data.

The two neural networks are trained with a version of the standard WGAN objective that was modified for *conditional* WGANs:

$$L = \mathop{\mathbb{E}}_{\mathbf{v} \sim P_v} \left(\mathop{\mathbb{E}}_{\tilde{\mathbf{x}} \in p_{\tilde{x}}(\mathbf{v})} [D(\tilde{\mathbf{x}}, \mathbf{v})] - \mathop{\mathbb{E}}_{\mathbf{x} \in p_x(\mathbf{v})} [D(\mathbf{x}, \mathbf{v})] \right) \tag{1}$$

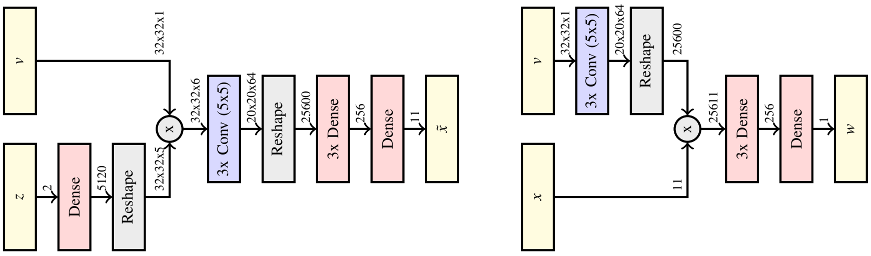

Fig. 2: Generator (left) and discriminator (right) model. All but the last layers use a Leaky ReLu activation. The generator also uses Batch normalization. The part of the generator output that represents the quaternion is normalized to have unit length and a Softmax over the 4 neurons that represent the grasp type is computed.

Here P_v is the distribution over all depth images in the training set, $p_x(v)$ is the distribution of successful grasps associated with an object depicted by depth image v and $p_{\tilde{x}}(v)$ is the distribution of generated grasps implicitly defined as $\tilde{x} = G(z, v), z \sim p(z)$. The generator tries to minimize the objective, whereas the discriminator tries to maximize it. Similar to [10], we try to enforce the Lipschitz constraint by adding a gradient penalty to the discriminator's loss:

$$L_{grad} = \mathop{\mathbb{E}}_{\mathbf{v} \sim P_v} \left(\mathop{\mathbb{E}}_{\hat{\mathbf{x}} \in p_{\hat{x}}(\mathbf{v})} \left[\left(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}}, \mathbf{v})\|_2 - 1 \right)^2 \right] \right) \tag{2}$$

Following [10], we implicitly define $p_{\hat{x}}(v)$ as sampling along straight lines between pairs of grasps sampled from the distribution of successful grasps $p_x(v)$ and the generator distribution $p_{\tilde{x}}(v)$: $\hat{x} = \alpha x + (1-\alpha)\tilde{x}$ with $\alpha \sim U(0,1)$. For training we used the same hyperparameters and training scheme as suggested by [10] and trained for 500 epochs. The model architecture is described in Figure 2.

3 Training Data

We train and evaluate our model on two different 3D object datasets. One comprising geometric primitives like boxes, spheres and cylinders and the other being the YCB dataset [11], which contains complex everyday objects. To generate grasp candidates we use two baseline methods.

3.1 Primitives

For the geometric primitives dataset, grasps were generated using a hand-engineered, deterministic grasping pipeline [12] that uses constrained superquadric fitting. Exploiting object symmetry, shape type, and object dimensions, a small set of grasp candidates is generated, which are ranked according to a hand-crafted heuristic. Note, that for the purpose of generating training data, we can skip the superquadrics fitting and use the geometric object properties from ground-truth.

Type	**Baseline**	**Baseline with fitting**	**GAN**
Box	98.6	78.9	98.3
Cylinder	88.7	72.0	85.6
Sphere	93.1	85.1	87.6
Total	93.5	78.7	90.5

Table 1: Grasp success ratios for our GAN approach compared to the baseline.

The training set comprises 6608 boxes, spheres, and cylinders of various sizes. For each object, the two highest-ranked grasps were considered for the training set. For each object-grasp pair, 16 rotations uniformly distributed over $[-\pi, \pi]$ around the z-axis are included in the training dataset.

3.2 YCB Dataset

To generate grasp candidates for the YCB dataset, we employed the grasping pipeline proposed by ten Pas et al. [4], which samples a large number of grasps and returns the 5 highest-ranked grasps using a deep convolutional network to predict grasp success from an input point cloud. This pipeline yields grasps for a parallel-yaw gripper, which we transform into grasps for our multi-fingered Shadow Robot hand using the All-Finger-Precision prototype and applying a fixed relative transformation.

The YCB dataset [11] comprises 73 real-world objects of different categories. Similar to the first dataset 8 uniformly distributed rotations around the z-axis of each object-grasp pair are included. The dataset was randomly split into training and test set using a ratio of 80:20.

4 Results & Discussion

4.1 Grasp Success Evaluation

4.1.1 Primitives

The trained model was tested on a set of 3000 objects with 1000 objects from each category. We report grasping success as the percentage of how many of these objects could be successfully lifted in simulation using grasps randomly sampled from the generator network. We compare this to the grasping pipeline that was used to create the grasps for the training data as a baseline. Because unlike our GAN the baseline directly accesses the true object properties we also compare to the baseline with a preliminary primitive fitting step to determine the object properties. The grasp success ratios show that the grasps generated by the GAN are approximately on par in quality with the baseline and outperform the baseline on similar conditions.

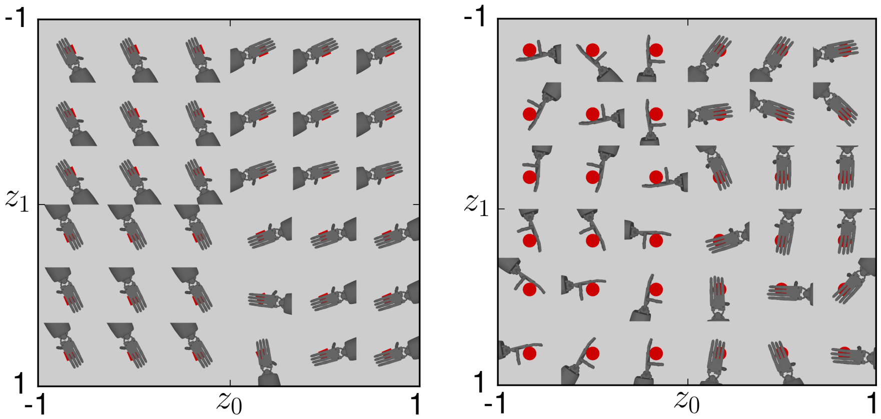

Fig. 3: Mapping from latent space to grasp space for a square box (left) and an upright cylinder (right).

4.1.2 YCB Dataset

On the YCB dataset, we compare our model against the discriminative approach proposed by ten Pas et al. [4]. As this approach predicts 5 grasps we also sample 5 grasps from our generator for a fair comparison. Our model achieves a best of 5 accuracy of 67.1% versus 68.75% accuracy for the discriminative approach. This suggests that our model can adapt to more complex objects and closely match the performance of discriminative approaches. Note that these accuracies are rather low. This is probably due to the conversion from two finger grasps to grasps with the artificial humanoid hand.

4.2 Latent Space Analysis

To analyze the learned mapping from latent space to grasp space, we simulate and render grasps for different inputs from the latent space for randomly selected primitive objects of different categories. The mappings are visualized in Figure 3.

Note that these mappings automatically emerged in an unsupervised fashion during training. In general, all mappings appear to have reoccurring characteristics that should be highlighted:

- The mapping from latent space to grasp space is mostly locally linear: Positions that are close in the latent space (and have the same sign) map to similar grasps. For example, for cylindric objects, a small movement in the latent space corresponds to a small change of the approach angle.

- Globally, the latent space partitions into several regions of semantically different grasp type – usually along a $z_i = 0$ axis. The space of successful grasps is not continuous and therefore also the mapping cannot be continuous and locally linear everywhere. It facilitates searching in the latent space that these discontinuities in the mapping correspond to a change of sign in the latent space.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

5 Conclusion

We present a conditional WGAN approach that is able to generate successful grasps based on a depth image input for primitive objects and objects from the YCB dataset. We compare the model against the grasp pipeline that was used for training data generation and show that it achieves the same grasp performance thus successfully mimicking the hand-crafted grasp generation process. Furthermore, it outperforms this baseline if the baseline uses primitive fitting instead of directly assessing the object's properties. On the YCB dataset, our model performs almost on par with a discriminative approach.

We further analyze the mapping from latent space to grasp space that was learned by the generator and find that useful properties for grasp selection have emerged. The mapping is locally linear and the latent space is usually ordered in a salient way for mappings: Different regions of the latent space map to distinguishable kinds of grasps.

References

[1] A. Makhal, F. Thomas, and A. P. Gracia, "Grasping unknown objects in clutter by superquadric representation," *arXiv preprint arXiv:1710.02121*, 2017.

[2] J. Mahler, S. Patil, B. Kehoe, J. Van Den Berg, M. Ciocarlie, P. Abbeel, and K. Goldberg, "Gp-gpis-opt: Grasp planning with shape uncertainty using gaussian process implicit surfaces and sequential convex programming," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4919–4926.

[3] D. Kappler, B. Bohg, and S. Schaal, "Leveraging big data for grasp planning," in *Proceedings of the IEEE International Conference on Robotics and Automation*, may 2015.

[4] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, p. 0278364917735594, 2017.

[5] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[6] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3406–3413.

[7] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesisa survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2014.

[8] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[9] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 5026–5033.

[10] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, 2017, pp. 5769–5779.

[11] B. Çalli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The YCB object and model set and benchmarking protocols," *CoRR*, vol. abs/1502.03143, 2015. [Online]. Available: http://arxiv.org/abs/1502.03143

[12] S. Meyer zu Borgsen, T. Korthals, F. Lier, and S. Wachsmuth, *ToBI Team of Bielefeld: Enhancing Robot Behaviors and the Role of Multi-Robotics in RoboCup@Home*. Springer, 2016, vol. 9776.