

Developing Creative AI to Generate Sculptural Objects

Songwei Ge*, Austin Dill*, Eunsu Kang*, Chun-Liang Li*, Lingyao Zhang
Manzil Zaheer, Barnabas Poczos

Carnegie Mellon University
Pittsburgh, PA, United States

{chunlial, eunsuk, songweig, lingyaoz, abdill, manzil, bapoczos}@andrew.cmu.edu

Abstract

We explore the intersection of human and machine creativity by generating sculptural objects through machine learning. This research raises questions about both the technical details of automatic art generation and the interaction between AI and people, as both artists and the audience of art. We introduce two algorithms for generating 3D point clouds and then discuss their actualization as sculpture and incorporation into a holistic art installation. Specifically, the Amalgamated DeepDream (ADD) algorithm solves the sparsity problem caused by the naive DeepDream-inspired approach and generates creative and printable point clouds. The Partitioned DeepDream (PDD) algorithm further allows us to explore more diverse 3D object creation by combining point cloud clustering algorithms and ADD.

Keywords

Partitioned DeepDream, Amalgamated DeepDream, 3D, Point Cloud, Sculpture, Art, Interactive Installation, Creative AI, Machine Learning

Introduction

Will Artificial Intelligence (AI) replace human artists or will it show us a new perspective into creativity? Our team of artists and AI researchers explore artistic expression using Machine Learning (ML) and design creative ML algorithms to be possible co-creators for human artists.

In terms of AI-generated and AI-enabled visual artwork, there has been a good amount of exploration done over the past three years in the 2D image area traditionally belonging to the realm of painting. Meanwhile, there has been very little exploration in the area of 3D objects, which traditionally would belong to the realm of sculpture and could be easily extended into the area of art installations. The creative generation of 3D object research by Lehman et al. successfully generated “evolved” forms, however, the final form was not far from the original form and could be said to merely mimic the original [22]. Another relevant study, Neural 3D Mesh Renderer [20] focuses on adjusting the rendered mesh based on DeepDream [28] textures. In the art field, artist Egor Kraft’s Content Aware Studies project [11] explores the possibilities of AI to reconstruct lost parts of antique Greek and Roman

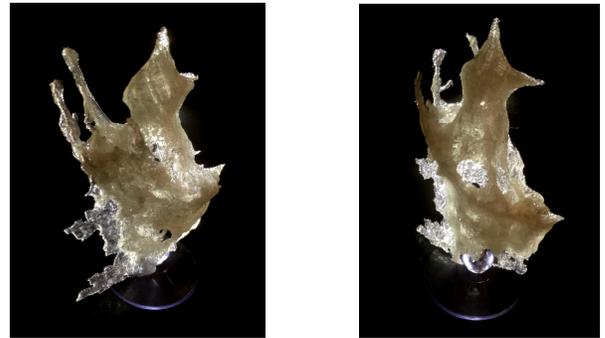


Figure 1: Sculpture generated by creative AI, PDD.

sculptures. Nima et al. [27] introduced Vox2Net adapted from pix2pix GAN which can extract the abstract shape of the input sculpture. DeepCloud [5] exploited an autoencoder to generate point clouds, and constructed a web interface with analog input devices for users to interact with the generation process. Similar to the work of Lehman et al [22], the results generated by DeepCloud are limited to previously seen objects.

In this paper, we introduce two ML algorithms to create original sculptural objects. As opposed to previous methods, our results do not mimic any given forms and are original and creative enough not to be categorized into the dataset categories. Our method extends an idea inspired by DeepDream for 2D images to 3D point clouds. However, simply translating the method generates low-quality objects with local and global sparsity issues that undermine the integrity of reconstruction by 3D printing. Instead we propose Amalgamated DeepDream (ADD) which utilizes union operations during the process to generate both creative and realizable objects. Furthermore, we also designed another ML generation algorithm, Partitioned DeepDream (PDD), to create more diverse objects, which allows multiple transformations to happen on a single object. With the aid of mesh generation software and 3D printing technology, the generated objects can be physically realized. In our latest artwork, it has been incorporated into an interactive art installation.

Creative AI: ADD and PDD

Artistic images created by AI algorithms have drawn great attention from artists. AI algorithms are also attractive for

*Equal contribution

their ability to generate 3D objects which can assist the process of sculpture creation. In graphics, there are many methods to represent a 3D object such as mesh, voxel and point cloud. In our paper, we focus on point cloud data which are obtained from two large-scale 3D CAD model datasets: ModelNet40 [38] and ShapeNet [9]. These datasets are preprocessed by uniformly sampling from the surface of the CAD models to attain the desired number of points. Point cloud is a compact way to represent 3D object using a set of points on the external surfaces and their coordinates such as those shown in Figures 2 and 3. We introduce two algorithms, ADD and PDD, which are inspired by 2D DeepDream to generate creative and realizable point clouds. In this section, we will briefly discuss the existing 3D object generation methods, and then elaborate the ADD and PDD algorithms and present some of the generated results.



Figure 2: ModelNet40 [38] Figure 3: ShapeNet [9]

Learning to Generate Creative Point Clouds

Using deep learning to generate new objects has been studied in different data types, such as music [35], images [15], 3D voxels [39] and point clouds [1, 23]. An example of a simple generative model is shown in Figure 4.

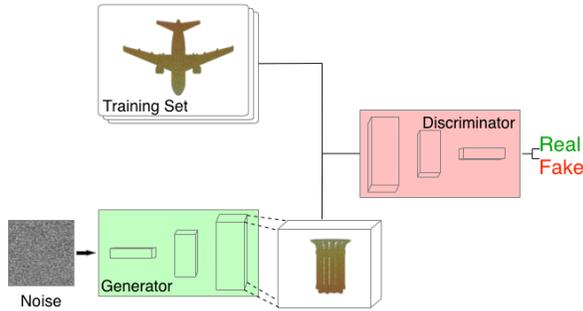


Figure 4: An example of generative model.

Here a low-dimensional latent space h of the original object space x is learned in accordance with the underlying probability distribution.

$$p(x) = \int_h p(h)p(x|h)dh \quad (1)$$

A discriminator is usually used to help distinguish the generated object \hat{x} and a real object x . Once the discriminator is fooled, the generator can create objects that look very similar to the original ones [15]. However, these generative models only learn to generate examples from the “same” distribution of the given training data, instead of learning to generate “creative” objects [13, 27].

One alternative is to decode the convex combination \hat{h} of latent codes h_1, h_2 of two objects in an autoencoder to get

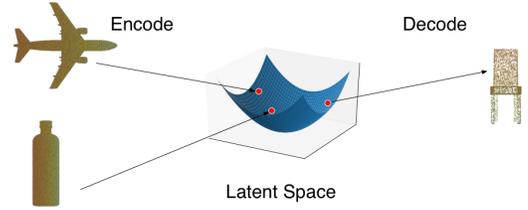


Figure 5: Encoding and decoding in latent space

final object \hat{X} . Specifically, an autoencoder is a kind of feedforward neural network used for dimensionality reduction whose hidden units can be viewed as latent codes which capture the most important aspects of the object [30]. A representation of the process of encoding to and decoding from latent space can be seen in Figure 5.

$$\hat{h} = t \cdot h_1 + (1 - t) \cdot h_2$$

$$\hat{x} = \text{Generator}(\hat{h}),$$

where $0 \leq t \leq 1$. Empirical evidence shows that decoding mixed codes usually produces semantically meaningful objects with features from the corresponding objects. This approach has also been applied to image creation [7]. One could adopt encoding-decoding algorithms for point clouds [40, 17, 23] based on the same idea. The sampling and interpolation results based on Li et al. [23] are shown in Figure 6.

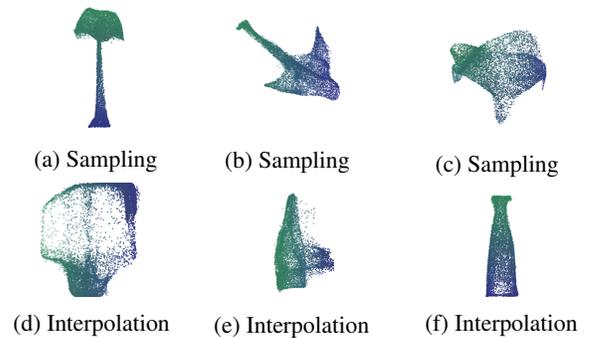


Figure 6: Results of Li et al. [23].

Inspiration: DeepDream for Point Clouds In contrast to simple generative models and interpolations, DeepDream leverages trained deep neural networks by enhancing patterns to create psychedelic and surreal images. For example in Figure 7, the neural network detects features in the input images similar to the object classes of bird, fish and dog and then exaggerates these underlying features. Given a trained neural network f_θ and an input image x , DeepDream aims to modify x to maximize (amplify) $f_\theta(x; a)$, where a is an activation function of f_θ . After this process, x is expected to display

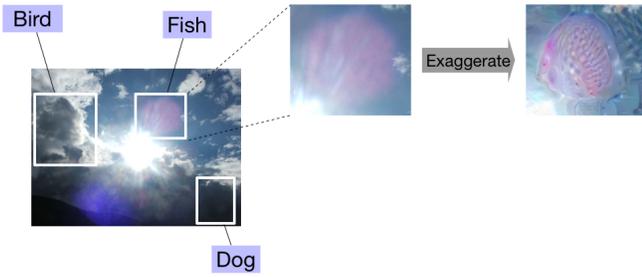


Figure 7: 2D DeepDream visualized

some features that are captured by $f_\theta(x; a)$. Algorithmically, DeepDream iteratively modifies x via a gradient update with a certain learning rate γ .

$$x_t = x_{t-1} + \gamma \nabla_x f_\theta(x; a). \quad (2)$$

An extension of DeepDream is to use a classification network as f_θ and we replace a with the outputs of the final layer corresponding to certain labels. This is related to adversarial attack [33] and unrestricted adversarial examples [6, 24].

Unfortunately, directly extending the idea of DeepDream to point clouds with neural networks for sets [29, 41] results in undesirable point clouds suffering from both local and global sparsity as shown in Figure 8. To be specific, iteratively applying gradient update without any restrictions creates local holes in the surface of the object. Besides, the number of input points per object is limited by the classification model. Consequently the generated object is not globally dense enough to transform into mesh structure, let alone realize in the physical world.

To avoid local sparsity in the generated point cloud, one compromise is to run the naive 3D point cloud DeepDream with fewer iterations or a smaller learning rate, but it results in limited difference from the input point cloud. Another approach to solving the sparsity problem is conducting post-processing to add points to sparse areas. However, without prior knowledge of the geometry of the objects, making smooth post-processed results is non-trivial [19]. For this reason, we simply take inspiration from the gradient update strategy of DeepDream, while developing our novel algorithms.

Amalgamated DeepDream (ADD)

As opposed to images, mixing two point clouds is a surprisingly simple task, which can be done by taking the union of two point clouds because of their permutation invariance property. For example, the union of two or three objects is shown in figure 9. This observation inspires us to use a set-union operation, which we call amalgamation, in the creation process to address the drawbacks of naive extension.

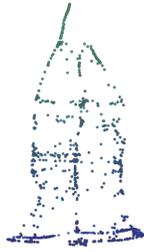
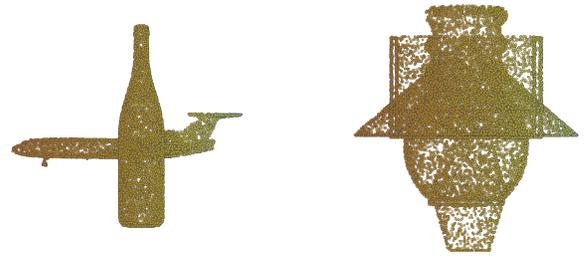


Figure 8: Naive DeepDream with undesirable sparse surface.



(a) Union of airplane and bottle (b) Union of cone, vase and lamp

Figure 9: Amalgamated input for ADD.

Algorithm 1: Amalgamated DeepDream (ADD)

```

input : trained  $f_\theta$  and input  $X$ 
output: generated object  $\hat{X}$ 
for  $x = X_0, \dots, X_n$  do
   $x_0 = x$ 
  for  $t = 1 \dots T$  do
     $\hat{x} = x_{t-1} + \gamma \nabla_x f_\theta(x_{t-1}; a)$ 
     $x_t = \hat{x}$ 
    if  $t$  is divisible by 10 then
       $x_t = x_t \cup x$ 
      down-sample  $x_t$  to fix the number of points
    end
  end
   $\hat{X} = \hat{X} \cup x_T$ 
end

```

For object X with any number of points, we first randomly take it apart into several subsets with the number of points that is required by the classification model, and then modify the input image through gradient updating (2). In the end, we amalgamate all the transformed subsets into a final object. By doing this we can generate as dense objects as we want as long as the input object allows. To solve the local sparsity, when running the gradient update (2) for each subset we amalgamate the transformed point clouds with the input point clouds after every 10 iterations. To avoid exploded number of points, we also down-sample the object after each amalgamation operation. We call the proposed algorithm Amalgamated DeepDream (ADD) as shown in Algorithm 1.

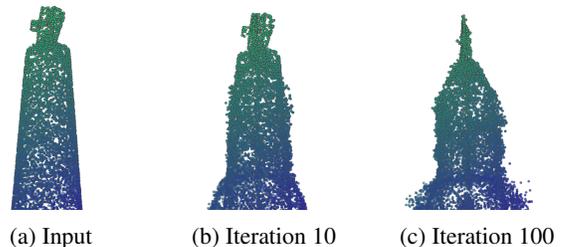
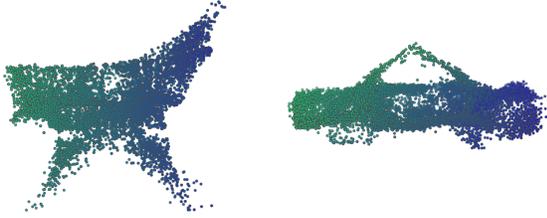


Figure 10: Transforming a bottle into a cone via ADD.

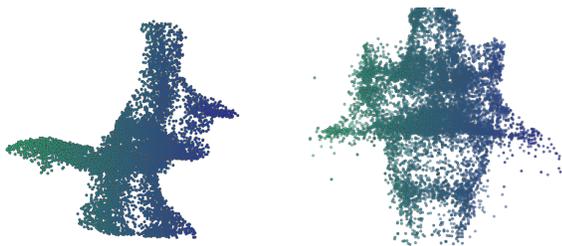


(a) Input: bowl; Target: chair (b) Input: keyboard; Target: car

Figure 11: ADD with single object input.

Experiments and results To test our methods, we use DeepSet [41] with 1000 points capacity as our basic 3D deep neural network model. We randomly sampled 10000 points from the CAD models and feed them into ADD model for 100 iterations with learning rate set to 1. A running example of ADD targeting the transformation of bottle into cone in ModelNet40 [38], which is the same as Figure 8, is shown in Figure 10. During the transformation process, ADD enjoys the advantage of deforming objects based on gradient updates with respect to a trained neural network as DeepDream without creating obviously local sparse areas. It can also better preserve the features of input point clouds with the amalgamation operation when we create new features based on DeepDream updates. In addition, the amalgamation of several transformed subsets from original objects allows us to generate denser objects that are realizable in the physical world. More created point clouds from the objects in Figure 2 are shown in Figure 11.

ADD with amalgamated inputs In addition to using union during the transformation process, we can push this idea further by using amalgamated point clouds as input instead of a single object. We keep the experimental setting the same. Note that now we could have multiple of 10000 points as input. The results of ADD with the union objects in Figure 9 are shown in Figure 12. Compared with Figure 11, ADD with multiple objects as input results in objects with more geometric details benefited from a more versatile input space.



(a) Input: airplane, bottle; Target: toilet (b) Input: cone, vase, lamp; Target: person

Figure 12: ADD with dual and triple object input.



Figure 13: Segmentation results of an airplane with 4 clusters.

Partitioned DeepDream (PDD)

As the example shown in Figure 7, applying DeepDream to images produces more than one surreal patterns on each of the images [28]. In contrast, the nature of ADD is to deform the whole object simultaneously into a certain shape. It is essential to extend such a basic algorithm in order to allow multiple, separate transformations on a single object. Therefore, we propose Partitioned DeepDream as shown in Algorithm 2 which can be implemented based on various point cloud segmentation algorithms. When compared with ADD, PDD provides a more controllable method for artists to explore with and effectively increases the diversity of creation.

Algorithm 2: Partitioned DeepDream (PDD)

input : trained f_θ , number of segments k , input X and targets a
output: generated object \hat{X}
 $S = \text{PCSegmentation}(X, k)$
for $x = S_1, \dots, S_k$ **do**
 Standardize x by $x = (x - \bar{x})/\sigma_x$
 $\hat{x} = \text{ADD}(f_\theta, x)$
 Recover \hat{x} by $\hat{x} = \hat{x} * \sigma_x + \bar{x}$
 $\hat{X} = \hat{X} \cup \hat{x}$
end

In contrast to the simplicity of mixing two point clouds, segmentation of point clouds is relatively hard and has been studied in depth [16, 26, 34]. Mainstream segmentation algorithms can be categorized into edge-based segmentation [31, 37], region growing segmentation [4], model-fitting segmentation [2, 14], hybrid segmentation [36] and machine learning segmentation [25]. In our work, manual segmentation is shown to obtain high-quality results but is extremely tedious and delegates too much responsibility to the artist.



(a) Input: airplane (b) Input: chair

Figure 14: PDD targeting on random classes

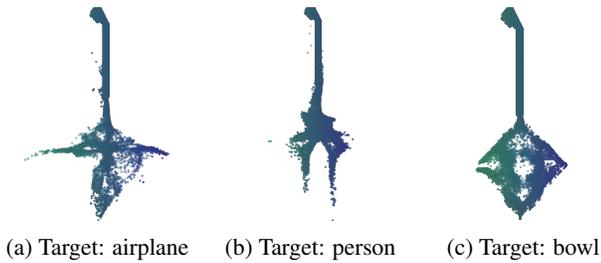


Figure 15: PDD with guitar body only

Instead, we explore machine learning methods which can automatically divide the object into several meaningful parts. For example, the segmentation results of an airplane with k -means are displayed in Figure 13. Note that this method requires the number of segments as input.

We extend ADD explained in Algorithm 1 with the k -means point cloud segmentation algorithm in our experiments. First, the input object is automatically divided into several segments using k -means. Each segment must be standardized before it is fed into ADD, in order for it to be treated as one distinct object. With this preprocessing step ADD allows us to produce the desired features on different parts independently. In the end, we cancel the normalization and reunion all segments into one object.

Experiments and results The same data preprocessing methods and ADD model configurations are used here as mentioned in the aforementioned ADD experiments. As for the partition strategies, one can randomly select some targeted categories for different parts and PDD results are displayed in Figure 14. This process is totally automatic with little human involvement. However, the random factors would lead to some undesired scattered points. Another more reasonable method is to create novel objects with more human participation by deforming the separate segments by design. As shown in Figure 15, we create some point clouds from a guitar shown in Figure 3 by only hallucinating on their body parts, while keeping the neck part intact. As we can see, the created objects are only locally modified while the over lay-

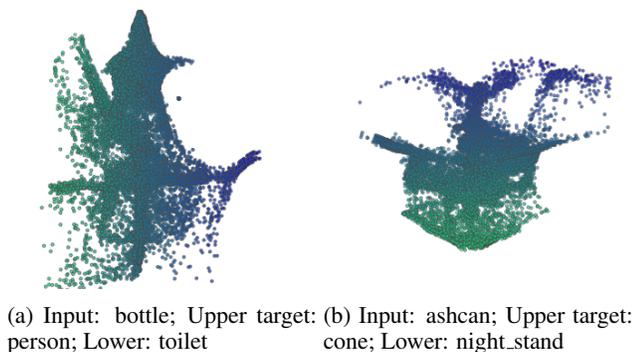


Figure 16: PDD on uniformly segmented objects

out remains that of a guitar.

PDD with manual segmentation strategies In addition to automatic segmentation algorithms, two manual segmentation strategies have been tested. First, for highly symmetric objects it is reasonable to uniformly divide it into several parts. As shown in Figure 16, we manually partition a bottle and an ashcan from their middle, and run PDD with some selected targets on two segments separately.

Another intuitive way to segment point clouds is to uniformly segment the whole spaces into many cubic blocks and treat the points in each block separately. The results based on this strategy are presented in Figure 17 and 18. We partition the whole space containing the airplane into $3 \times 3 \times 2$ blocks alongside three axes, and generate features on each block with either a random class or cone class using PDD. In comparison to ADD, PDD can generate more delicate objects since it can manipulate different units of a single object. Besides, this also indicates a more expansive space for generating creative 3D objects.



Figure 17: PDD with block segmented objects targeting random categories. Figure 18: PDD with block segmented objects targeting cone.

Realization of Sculptural Object

From points clouds to mesh Given a novel point cloud, our task becomes generating a mesh – a description of an object using vertices, edges, and faces – in order to eliminate the spatial ambiguity of a point cloud representation. While mesh generation, also known as surface reconstruction, is readily provided in the free software package MeshLab [10], each possible algorithm has its own use case, so this task involves experimenting with each available algorithm to find what generates the best reconstruction for a given point cloud.

Our experiments focused on the following reconstruction algorithms: Delaunay Triangulation [8], Screened Poisson [21], Alpha Complex [18], and Ball-Pivoting [3]. While each of these algorithms has its uses, ADD results in point clouds with a greater variance in sparsity than typical point clouds generated from real-world objects.

For all experiments, we limited the number of points to 10,000. If a generated point cloud had more than this number, we reduced using Poisson Disk Sub-sampling [12]. This sub-sampling method is more intelligent than a simple uniform resampling and produces points that appear to balance sparsity with finer definition.

An illustration of our findings can be found in Figure 19. Typically, Delaunay Triangulation produces a mesh that eliminates all detail of the point cloud. This approach will therefore only be appropriate when it is applied to an exceptionally sparse point cloud, as it would contain minimal detail

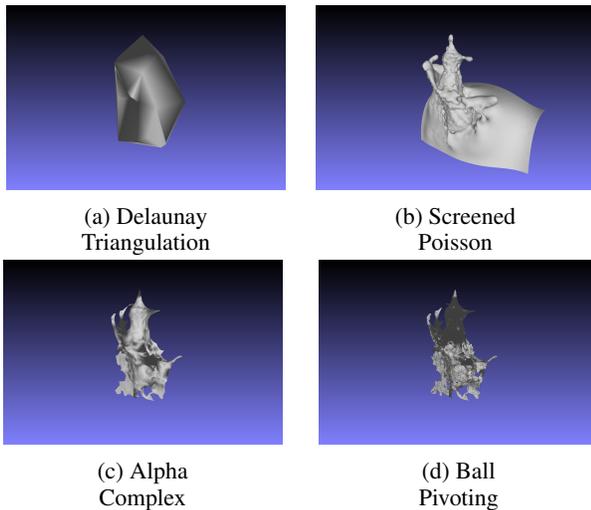


Figure 19: Reconstruction techniques

regardless of the reconstruction technique. Screened Poisson extrapolates from the original point cloud, producing interesting shapes that are unfortunately misleading of the underlying structure. Alpha Complex and Ball-Pivoting are the two most consistent reconstruction methods and they both produce similar meshes. One important difference is that Alpha Complex produces non-manifold edges, all but making faithful 3D printing impossible.

As a result of these trials, we settled on Ball-Pivoting after Poisson Disk Sub-sampling to produce our final mesh.

From mesh to realizable 3D sculptures Our final goal is to create 3D sculptures in the real world. We use standard software Meshmixer [32] to solidify the surfaced reconstructed point cloud created by ADD. The solidified version of Figure 19d can be seen in Figure 1. We then use Lulzbot TAZ 3D printer with dual-extruder, transparent PLA, and dissolvable supporting material to create the sculpture of the reconstructed mesh. The printed sculptures of the point cloud in Figure 12 (RHS) are shown in Figure 21.

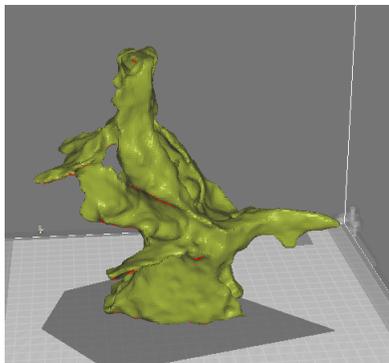


Figure 20: Screen capture from Cura-Lulzbot software for 3D printing



Figure 21: Created sculpture from ADD.



Figure 22: Sketch of Aural Fauna: Illuminato art installation and a Photo from the touch-sound interaction test with tablet computer interface

Conclusion

This paper describes our development of two ML algorithms for point cloud generation, ADD and PDD, which possess the ability to generate unique sculptural artwork, distinct from the dataset used to train the underlying model. These approaches allow more flexibility in the deformation in the underlying object as well as the ability to control the density of points required for meshing and printing 3D sculptures.

Through our development of ADD and PDD for sculptural object generation, we show AI’s potential for creative endeavors in art, especially the traditional field of sculpture and installation art. One challenge in the development of creative AI is the contradiction between the need of an objective metric for evaluating the quality of generated objects in the ML literature and the desire for creative and expressive forms in the art community. Reconciling these goals will require a mathematical definition of creativity that can help guide the AI’s generative process for each specific case. We will undertake these deep computational and philosophical questions in future work.

In the meantime, the realized 3D objects are an integral part of our ongoing project that is an interactive art installation. This project, Aural Fauna: Illuminato, presents an unknown form of organism, aural fauna, that reacts to the visitors sound and touch. Their sound is generated using machine learning. The creatures bodies are the generated sculptural objects by ADD and PDD. The sculptural objects could be installed on the wall or be hung from the ceiling like a swarm and the participant may interact with them using the touch-screen interface and their voice input as seen in Figure 22. This project aims to encapsulate the joint efforts of human artists and Creative AI as well as the viewer as partners in the endeavor.

References

[1] Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; and Guibas, L. 2017. Learning representations and generative models

- for 3d point clouds. *arXiv preprint arXiv:1707.02392*.
- [2] Ballard, D. H. 1981. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition* 13(2):111–122.
- [3] Bernardini, F.; Mittleman, J.; Rushmeier, H.; Silva, C.; and Taubin, G. 1999. The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics* 5(4):349–359.
- [4] Besl, P. J., and Jain, R. C. 1988. Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10(2):167–192.
- [5] Bidgoli, A., and Veloso, P. 2018. Deepcloud: The application of a data-driven, generative model in design.
- [6] Brown, T. B.; Carlini, N.; Zhang, C.; Olsson, C.; Christiano, P.; and Goodfellow, I. 2018. Unrestricted adversarial examples. *arXiv preprint arXiv:1809.08352*.
- [7] Carter, S., and Nielsen, M. 2017. Using artificial intelligence to augment human intelligence. *Distill*.
- [8] Cazals, F., and Giesen, J. 2006. Delaunay triangulation based surface reconstruction. In *Effective computational geometry for curves and surfaces*. Springer. 231–276.
- [9] Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.
- [10] Cignoni, P.; Callieri, M.; Corsini, M.; Dellepiane, M.; Ganovelli, F.; and Ranzuglia, G. 2008. MeshLab: an Open-Source Mesh Processing Tool. In Scarano, V.; Chiara, R. D.; and Erra, U., eds., *Eurographics Italian Chapter Conference*. The Eurographics Association.
- [11] Content aware studies. <http://egorkraft.art/#cas>.
- [12] Corsini, M.; Cignoni, P.; and Scopigno, R. 2012. Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE Transaction on Visualization and Computer Graphics* 18(6):914–924. <http://doi.ieeecomputersociety.org/10.1109/TVCG.2012.34>.
- [13] Elgammal, A.; Liu, B.; Elhoseiny, M.; and Mazzone, M. 2017. Can: Creative adversarial networks, generating” art” by learning about styles and deviating from style norms. *arXiv preprint arXiv:1706.07068*.
- [14] Fischler, M. A., and Bolles, R. C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24(6):381–395.
- [15] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*.
- [16] Grilli, E.; Menna, F.; and Remondino, F. 2017. A review of point clouds segmentation and classification algorithms. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 42:339.
- [17] Groueix, T.; Fisher, M.; Kim, V. G.; Russell, B. C.; and Aubry, M. 2018. Atlasnet: A papier-mache approach to learning 3d surface generation. *arXiv preprint arXiv:1802.05384*.
- [18] Guo, B.; Menon, J.; and Willette, B. 1997. Surface reconstruction using alpha shapes. In *Computer Graphics Forum*, volume 16, 177–190. Wiley Online Library.
- [19] Hoppe, H.; DeRose, T.; Duchamp, T.; McDonald, J.; and Stuetzle, W. 1992. *Surface reconstruction from unorganized points*.
- [20] Kato, H.; Ushiku, Y.; and Harada, T. 2018. Neural 3d mesh renderer. In *CVPR*.
- [21] Kazhdan, M., and Hoppe, H. 2013. Screened poisson surface reconstruction. *ACM Transactions on Graphics (TOG)* 32(3):29.
- [22] Lehman, J.; Risi, S.; and Clune, J. 2016. Creative generation of 3d objects with deep learning and innovation engines. In *Proceedings of the 7th International Conference on Computational Creativity*.
- [23] Li, C.-L.; Zaheer, M.; Zhang, Y.; Poczos, B.; and Salakhutdinov, R. 2018. Point cloud gan. *arXiv preprint arXiv:1810.05795*.
- [24] Liu, H.-T. D.; Tao, M.; Li, C.-L.; Nowrouzezahrai, D.; and Jacobson, A. 2018. Adversarial geometry and lighting using a differentiable renderer. *arXiv preprint arXiv:1808.02651*.
- [25] Lu, X.; Yao, J.; Tu, J.; Li, K.; Li, L.; and Liu, Y. 2016. Pairwise linkage for point cloud segmentation. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences* 3(3).
- [26] Nguyen, A., and Le, B. 2013. 3d point cloud segmentation: A survey. In *RAM*, 225–230.
- [27] Nima, D.; Luca, S.; and Mauro, M. 2018. Vox2net: From 3d shapes to network sculptures. In *NIPS*.
- [28] Posted by Alexander Mordvintsev, C. O., and Tyka, M. Inceptionism: Going deeper into neural networks. <https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html>.
- [29] Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CVPR*.
- [30] Robert, C. 2014. Machine learning, a probabilistic perspective.
- [31] Sappa, A. D., and Devy, M. 2001. Fast range image segmentation by an edge detection strategy. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, 292–299. IEEE.
- [32] Schmidt, R., and Singh, K. 2010. Meshmixer: an interface for rapid mesh composition. In *ACM SIGGRAPH 2010 Talks*, 6. ACM.
- [33] Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

- [34] Trevor, A. J.; Gedikli, S.; Rusu, R. B.; and Christensen, H. I. 2013. Efficient organized point cloud segmentation with connected components. *Semantic Perception Mapping and Exploration (SPME)*.
- [35] Van Den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A. W.; and Kavukcuoglu, K. 2016. Wavenet: A generative model for raw audio. In *SSW*.
- [36] Vieira, M., and Shimada, K. 2005. Surface mesh segmentation and smooth surface extraction through region growing. *Computer aided geometric design* 22(8):771–792.
- [37] Wani, M. A., and Arabnia, H. R. 2003. Parallel edge-region-based segmentation algorithm targeted at reconfigurable multiring network. *The Journal of Supercomputing* 25(1):43–62.
- [38] Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*.
- [39] Wu, J.; Zhang, C.; Xue, T.; Freeman, B.; and Tenenbaum, J. 2016. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NIPS*.
- [40] Yang, Y.; Feng, C.; Shen, Y.; and Tian, D. 2018. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *CVPR*, volume 3.
- [41] Zaheer, M.; Kottur, S.; Ravanbakhsh, S.; Póczos, B.; Salakhutdinov, R. R.; and Smola, A. J. 2017. Deep sets. In *NIPS*.

Authors Biographies

Songwei Ge is a Masters student in the Computational Biology Department at Carnegie Mellon University.

Austin Dill is a Masters student in the Machine Learning Department at Carnegie Mellon University.

Chun-Liang Li is a PhD candidate in the Machine Learning Department of Carnegie Mellon University. He received IBM Ph.D. Fellowship in 2018 and was the Best Student Paper runner-up at the International Joint Conference on Artificial Intelligence (IJCAI) in 2017. His research interest is on deep generative models from theories to practical applications.

Dr. Eunsu Kang is a Korean media artist who creates interactive audiovisual installations and AI artworks. Her current research is focused on creative AI and artistic expressions generated by Machine Learning algorithms. Creating interdisciplinary projects, her signature has been seamless integration of art disciplines and innovative techniques. Her work has been invited to numerous places around the world including Korea, Japan, China, Switzerland, Sweden, France, Germany, and the US. All ten of her solo shows, consisting of individual or collaborative projects, were invited or awarded. She has won the Korean National Grant for Arts three times. Her researches have been presented at prestigious conferences including ACM, ICMC, ISEA, and

NeurIPS. Kang earned her Ph.D. in Digital Arts and Experimental Media from DXARTS at the University of Washington. She received an MA in Media Arts and Technology from UCSB and an MFA from the Ewha Womans University. She had been a tenured art professor at the University of Akron for nine years and is currently a Visiting Professor with emphasis on Art and Machine Learning at the School of Computer Science, Carnegie Mellon University.

Lingyao Zhang earned her Master’s degree in the Machine Learning Department at Carnegie Mellon University.

Manzil Zaheer earned his Ph.D. degree in Machine Learning from the School of Computer Science at Carnegie Mellon University under the able guidance of Prof Barnabas Póczos, Prof Ruslan Salakhutdinov, and Prof Alexander Smola. He is the winner of Oracle Fellowship in 2015. His research interests broadly lie in representation learning. He is interested in developing large-scale inference algorithms for representation learning, both discrete ones using graphical models and continuous with deep networks, for all kinds of data. He enjoys learning and implementing complicated statistical inference, data-parallelism, and algorithms in a simple way.

Dr. Barnabas Pczos is an associate professor in the Machine Learning Department at the School of Computer Science, Carnegie Mellon University. His research interests lie in the theoretical questions of statistics and their applications to machine learning. Currently he is developing machine learning methods for advancing automated discovery and efficient data processing in applied sciences including health-sciences, neuroscience, bioinformatics, cosmology, agriculture, robotics, civil engineering, and material sciences. His results have been published in top machine learning journals and conference proceedings, and he is the co-author of 100+ peer reviewed papers. He has been a PI or co-Investigator on 15+ federal and non-federal grants. Dr. Póczos is a member of the Auton Lab in the School of Computer Science. He is a recipient of the Yahoo! ACE award. In 2001 he earned his M.Sc. in applied mathematics at Eotvos Lorand University in Budapest, Hungary. In 2007 he obtained his Ph.D. in computer science from the same university. From 2007-2010 he was a postdoctoral fellow in the RLAI group at University of Alberta, then he moved to Pittsburgh where he was a postdoctoral fellow in the Auton Lab at Carnegie Mellon from 2010-2012.