Proceedings of the ASME 2019 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2019
August 18-21, 2019, Anaheim, CA, USA

IDETC2019-98525

3D SHAPE SYNTHESIS FOR CONCEPTUAL DESIGN AND OPTIMIZATION USING VARIATIONAL AUTOENCODERS

Wentai Zhang **Zhangsihao Yang** **Haoliang Jiang** **Suyash Nigam**
Soji Yamakawa **Tomotake Furuhata** **Kenji Shimada** **Levent Burak Kara**[*]
Carnegie Mellon University, Pittsburgh, PA, USA

ABSTRACT

We propose a data-driven 3D shape design method that can learn a generative model from a corpus of existing designs, and use this model to produce a wide range of new designs. The approach learns an encoding of the samples in the training corpus using an unsupervised variational autoencoder-decoder architecture, without the need for an explicit parametric representation of the original designs. To facilitate the generation of smooth final surfaces, we develop a 3D shape representation based on a distance transformation of the original 3D data, rather than using the commonly utilized binary voxel representation. Once established, the generator maps the latent space representations to the high-dimensional distance transformation fields, which are then automatically surfaced to produce 3D representations amenable to physics simulations or other objective function evaluation modules. We demonstrate our approach for the computational design of gliders that are optimized to attain prescribed performance scores. Our results show that when combined with genetic optimization, the proposed approach can generate a rich set of candidate concept designs that achieve prescribed functional goals, even when the original dataset has only a few or no solutions that achieve these goals.

INTRODUCTION

In engineering design, while design simulation and analysis technologies are well developed and ubiquitous, digital tools to assist the early conceptual design phases are severely limited. Instead, humans still play a critical role in establishing the design space and the associated parameterizations. However, the heavy reliance on human-driven concept generation and design space exploration make product development particularly challenging for problems in which the geometry/form of the product has a significant impact on performance. As such, the need for digital design tools that support (1) knowledge extraction from configurationally and geometrically different past designs, (2) leveraging this information for large-variance, automatic design synthesis inside and outside of the original design space, and (3) seamless integration into analysis and simulation engines remain a central need in design automation.

In this work, we present a data-driven 3D shape synthesis method to assist human designers in conceptual design. Our approach relies on the observation that past designs may encapsulate useful design information that, if digitally captured, could be used to generate new designs automatically. To this end, we adopt an unsupervised variational autoencoder (VAE) deep learning method that takes input a corpus of 3D designs and extracts a latent design representation. This representation transforms the originally very high-dimensional data into a compact feature vector, where each feature encodes a latent probability distribution function learned over all past designs. Once learned, this representation can be sampled, or elements in this latent space can be interpolated and extrapolated to generate new latent space instances. These new instances can then be projected to the original design space using the decoder of the VAE.

In contrast to the common method of voxelizing 3D data into a 3D binary representation [1–4], we utilize distance transformation maps [5] as the primary input. This involves (1) the conversion of input 3D shapes (commonly acquired in the form

[*] Address all correspondences to lkara@cmu.edu

Copyright © 2019 by ASME

of polygonal models) into real valued distance maps, (2) using the distance map of each original design to train the VAE, (3) automatically converting any synthetically generated new distance map back to a polygonal model for downstream analysis. This allows the synthesized 3D shapes to exhibit much smoother surfaces without suffering from 'pixelatization,' while being amenable to engineering analyses.

We demonstrate the utility of our approach on the 3D outer shape design of gliders. While one approach is to learn a direct mapping from the available past designs to their aerodynamic performance, and use this mapping as a simulator to evaluate new designs, such a mapping would need to be learned for every new engineering objective. Instead, VAE learns a shape generator in an unsupervised manner, where the latent space exploration in the trained architecture allows the newly generated design to be integrated into widely available analysis tools. We demonstrate this in a particular case study, where the study incorporates flight dynamics (albeit simplified) for shape optimization.

Our approach also develops a simple latent space design crossover technique that allows a genetic optimizer to produce a large set of new designs through stochastic latent vector interpolation and extrapolation. While not a requirement in the overall framework, the use of genetic optimization enables a large set of synthetically generated designs to meet the target performance score, even though only a few or no original design solutions could attain the prescribed goals. This, in turn, offers greater conceptual latitude to the human designer in deciding which concepts to further develop.

BACKGROUND

With advances in machine learning, data-driven approaches that model and optimize engineering design problems are becoming increasingly more prevalent. In this section, we will discuss some recent progress in (1) addressing various mechanical problems with data-driven approaches, (2) applications of learning-based methods in design and (3) dimensionality reduction for shape optimization.

Data-driven Methods in Various Mechanical Problems

Sosnovik et al. [6] propose a Convolutional Neural Network that accelerates topology optimization computations. Using the powerful ability of deep learning methods to segment images pixel-wise, the approach predicts final optimal topologies after several iterations of optimization, based upon the initial conditions of the layout.

To address the circuit synthesis in EDA design, Guo et al. [7] propose an active learning strategy for reducing topology evaluation cost for a circuit synthesis problem. They utilize a predictive model with a random forest to approximate true circuit topology performance. Their experimentation reveals that uncertainty and topology structure may play critical roles in improving the appropriation model accuracy and make a significant contribution to reducing the system evaluation costs.

Data-driven methods have also been used in modeling nonlinear physics. Raissi et al. [8] treats the deep neural network as a non-linear function approximator and use the method to identify complex non-linear systems such as Lorenz system and the Glycoltic oscillator model. Umetani et al. [9] develop a data-driven approach to estimate the aerodynamic forces on a glider and its wing shape, and use this for glider design. This enables a user to accurately match the desired trajectory without the aid of costly simulations or experiments.

Learning-based Methods in Design Problems

With a large amount of available data and the advance of hardware technology, data-driven methods have become an increasingly common strategy for the problems that are difficult to approach by the creating physical model or are expensive in computation. Recently, researchers in the mechanical design community have started exploring the machine learning approach to aid the design process.

Fuge et al. [10] devise a framework that relies on collaborative filtering to recommend best design methodologies to solve target design problems, and argue that such approaches can be valuable for novice designers and enhance the overall product development cycle. In order to have an automatic design generator, Chen et al. [11] introduce the BezierGAN, a generative model for synthesizing smooth curves. The model maps a low-dimensional latent representation to a sequence of discrete points sampled from a rational Bezier curve. It is tested on four different design datasets and reveals better capacities in generating realistic 2D smooth shapes when compared with InfoGAN. Similar frameworks can have an impressive performance even for creative hand sketches. Chen et al. [12] propose a model, sketch-pix2seq, based on a sequence-to-sequence variational-auto-encoder (VAE) model called sketch-rnn. With their modification, the model has better performance in learning and generating sketches of multiple categories and shows promising results in creative tasks. However, these works are limited to 2D images or sketches. It is still challenging to extend these works to 3D design representations. Unlike 2d images which can be easily represented as unified RGB/grayscale form with the same size, 3d shapes are usually represented in polygonal meshes, point clouds or implicit surfaces with various lengths and weak correspondence among different shapes.

To have a deeper understanding of the mapping between the shape and the functions of design, Dering et al. [13] propose a deep learning approach based on three-dimensional (3D) convolutions that predict functional quantities of digital design concepts. Testing trained models on novel input yields accuracy as high as 98% for estimating the rank of the functional quantities.

Copyright © 2019 by ASME

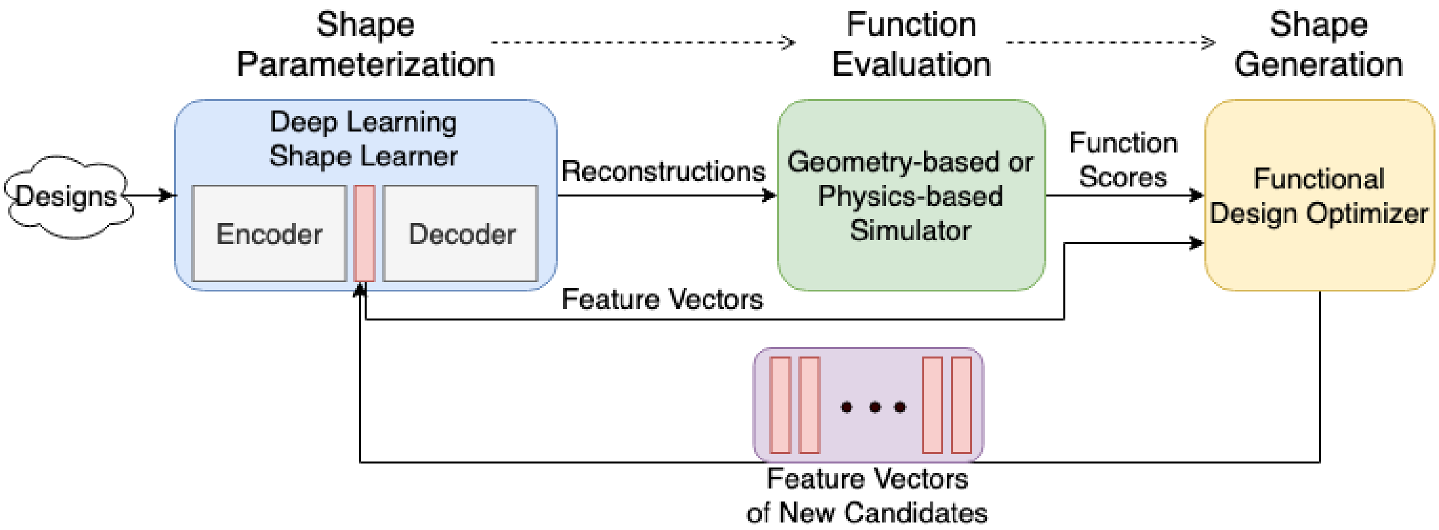
Shape Parameterization
Function Evaluation
Shape Generation
Designs
Deep Learning Shape Learner
Encoder
Decoder
Reconstructions
Geometry-based or Physics-based Simulator
Function Scores
Functional Design Optimizer
Feature Vectors
Feature Vectors of New Candidates

FIGURE 1. The architecture of our proposed data-driven conceptual design pipeline.

This method is also employed to differentiate between decorative and functional headwear. Moreover, Burnap et al. [14] develop a deep learning approach to predict design gaps in the market. Their approach is built on conventions in both quantitative marketing inbounding the heterogeneity of consumer choice preferences, as well as engineering design for bounding the space of possible designs. Raina et al. [15] explore the representation of design strategies as a Hidden Markov Model and their application to engineering design problems. Their results imply the successful transfer of design strategies from human designers to computational agents. They also propose a method to achieve transfer learning in agent-based models through state-based probabilistic models. Burnap et al. [16] use a deep learning based generative model to find the statistical representation of a design space via using a large number of images and design attributes. They test their methods on automobile body design and successfully morph a vehicle into different meaningful body types. In consideration of a sequential design pipeline, Oh et al. [17] manage to combine the generative methods with further topology optimization in automobile wheel design. However, the synthetic designs are still technically immature with one-shot optimization. They claim that an iterative and automatic optimization process can be a better alternative.

Dimensionality Reduction for Shape Optimization

At the heart of a functional shape design process, the idea of reducing the dimension in the original design space plays a critical role in shape optimization. For example, generative topographic mapping has been proposed by Viswanath et al. [18, 19] to tackle with the design problem of 2D aircraft wings. In order to reduce the dimension of design space with a sense of the shape common features, Proper Orthogonal Decomposition (POD) [20] and Generative Adversarial Networks (GAN) [21] are introduced as effective shape feature extractors. They both validate their design framework in designing a 2D airfoil profile through limited control points.

Although the aforementioned prior researches achieve plausible results when applying a data-driven approach as a tool for the design process in 2D design space, there doesn't exist a close-loop 3D conceptual design pipeline which enhances the design automation and optimization with respect to the functional requirements.

TECHNICAL APPROACH

Fig. 1 shows our proposed framework. The key modules are a deep learned shape encoder-decoder, a geometry or physics-based design simulator, and an optimizer.

Input to our approach is a database of 3D models belonging to the same object category (e.g., aircrafts). These models are most commonly acquired in the form of 3D polygonal surface models. Through unsupervised learning, the variational shape learner extracts a latent feature vector for each of the input designs. The latent space vector has much fewer dimensions than the original shape representation, and thus serves as a dimension reduced encoding of the large design space. This is the first module of our design pipeline (Shape Parameterization).

The geometry- or physics-based simulator is determined by the design performance objectives. It is responsible for testing the design candidates and provide performance scores for the subsequent optimization process (Function Evaluation). The inputs of this module are the reconstructions of some candidate feature vectors using the existing well-trained decoder network.

Copyright © 2019 by ASME

Through evaluation, we can have all the function scores for these design candidates, which aid the optimizer to generate potential better candidates.

The optimizer utilizes the outcomes of the simulator, together with the latent space representation, to optimize the designs directly in the latent space. New candidates can be generated with a non-gradient based optimization method (Shape Generation). These modules are detailed further in the following sections.

Variational Shape Learner

In this study, we adopt a variational autoencoder model, the Variational Shape Learner (VSL) [22], which builds on the ideas of the Neural Statistician [23] and the volumetric convolutional network [24]. The parameters of the VSL are learned under a variational inference scheme [25]. As shown in Fig. 2, we use a hierarchical VAE (Variational Autoencoder) which consists of an encoder, a decoder and a latent space feature representation.

Distance Maps and Shape Representation: We use a 3D signed distance field (SDF) as the primary design representation in the original space. As opposed to the commonly used binary voxel representation [1–4], this representation allows smooth final surfaces to be constructed over the designs generated by VSL.

SDF is a scalar function of position that defines a closed volume implicitly. The absolute value of the function is the distance from the surface of the solid. A positive value indicates the point is inside of the solid, or negative outside. The boundary is the isosurface where the function value is zero. SDF has been recently used for deep-learned shape completions [5].

In our approach, the SDF is implemented as a tri-linear function defined over a structured lattice. A signed distance is assigned to the lattice nodes and interpolated within the lattice cell. A lattice with $n \times m \times k$ nodes has $(n-1) \times (m-1) \times (k-1)$ cells, and each node is connected to neighboring nodes by multiple edges.

To create the training and testing data, an SDF needs to be generated from a polygonal mesh. The program calculates a distance d from each lattice node to the nearest point on the polygonal mesh. Then $+d$ is assigned to the node if the node is inside of the solid, or $-d$ if outside.

Once an SDF is obtained as an output from the VSL, a polygonal mesh of the boundary needs to be extracted from the SDF for rendering and to facilitate downstream processes.

For this, our approach first places a vertex on each edge with the two nodes having one positive and one negative value. The position of the vertex is smoothly interpolated based on the distance values.

Then the program creates arcs on each rectangle of the lattice cells that has a mix of positive and negative nodes. There are four cases: (1) one negative and three positive, (2) two negative connected and two positive connected, (3) two negative diagonal and the other two positive, and (4) one positive and three negative. Cases (1), (2), and (4) yield one arc, and (3) yields two arcs as shown in Fig. 3.

Finally, polygonal faces in each lattice cell are created by connecting the arcs. The result may become degenerate if a lattice node has exactly zero value, or the node is exactly on the boundary. We append a very small value to the distance values of such nodes to avoid this degeneracy.

Encoder-Decoder Details: For the encoder, the global latent code is directly learned from the input SDF through three convolutional layers with kernel sizes 6, 5, 4, strides 2, 2, 1 and channels 32, 64, 128. The size of the latent vector is 70. Each layer is followed by a ReLU activation layer [26] and a batch normalization layer [27]. Each local latent code is conditioned on the global latent code, the input voxel and the previous latent code (except for the first local latent code) using two fully-connected layers with 100 neurons each. After we learn the global and local latent codes, we concatenate them into a single vector. A 3D deconvolutional neural network with dimensions symmetrical to the encoder of the global latent code is used to decode the learned latent features into an output SDF model.

An element-wise logistic sigmoid [28] is applied to the output layer in order to convert the learned features to occupancy probabilities for each SDF cell. The detailed network architecture can be found in [22].

Physics-based Simulator

The trajectories of the original and synthesized aircrafts are simulated in YS FLIGHT SIMULATOR [29]. While the flight dynamics kernel of the simulator uses a simplified physical model, it is suitable for making a quick estimation of the flight characteristics.

An aircraft in the air is subject to lift, drag, gravity, and thrust, so called the four forces of flight. For this design task thrust is cut to idle, or zero, rendering the designs as gliders. Gravity is -9.8m/s^2 in the y-direction. Lift and drag forces are calculated as:

$$L = \frac{1}{2} C_L \rho v^2 S \tag{1}$$

$$D = \frac{1}{2} C_D \rho v^2 S, \tag{2}$$

where L is lift, C_L is lift coefficient, ρ is air density, v is velocity, S is wing area, D is drag, and C_D is drag coefficient. C_L and C_D are functions of α or angle-of-attack. YS FLIGHT SIMULATOR kernel approximates C_L and C_D as a piecewise-linear and a parabolic function respectively.

Copyright © 2019 by ASME

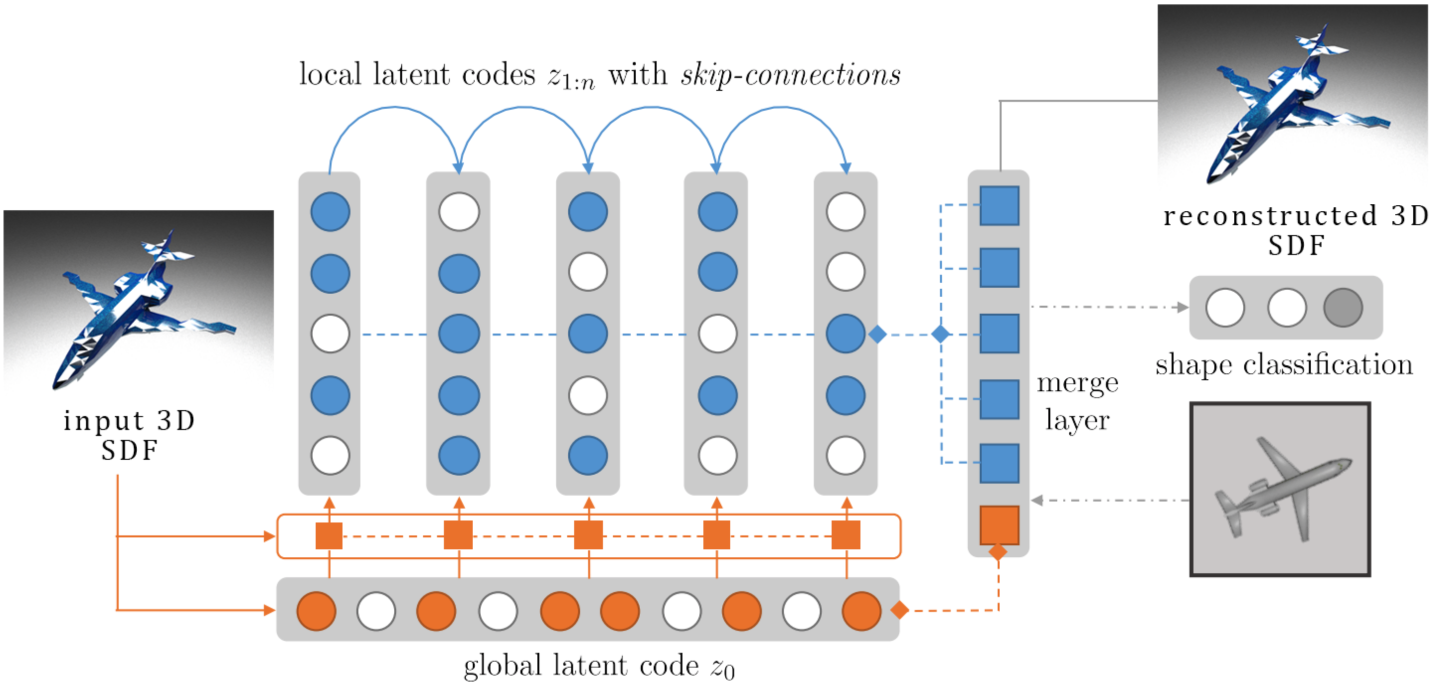

FIGURE 2. A schematic of VSL [22].

The rotation of an aircraft is the hardest to simulate. Unless the moment of inertia and center of gravity is known it is impossible to make an accurate simulation. Instead of estimating the moment of inertia and center of gravity, the simulator kernel approximates a rotation as a second-order system with a stability constant and a maneuverability constant, both of which are empirically specified.

Estimating a reasonable C_L and C_D functions and stability and maneuverability constants for an output design of the network is also a challenge. However, the output design from the network leaves freedom of choosing the airfoil of the wing, moment of inertia, and the center of gravity. It is reasonable to assume that the designs generated by the network can be configured to have a similar characteristic to an existing airplane of a similar geometric signature. We have chosen one from 88 aircraft data in the flight simulator that matches the forward-projection and top-view-projection area ratio the best to the output design from the network and take C_L and C_D functions and stability and maneuverability constants for the characteristic of the designs generated by the network.

Design Optimizer

The main purpose of this module is to explore potentially better design candidates using the compact latent space. We use a gradient-free genetic optimizer following its success in conceptual design [30–32]. Latent feature vectors extracted from various designs are selected and ranked based on their corresponding performance scores. Then, mutation and crossover operations involving interpolation and extrapolation are used to produce subsequent generations.

We observe that conventional crossover renders shapes incomplete. This means that not all the designs in the latent space are valid. To address this problem, we utilize the line crossover operator [33] to generate new offsprings. A child is generated using a linear interpolation between two parents:

$$Child = r \times P_1 + (1-r) \times P_2, \tag{3}$$

where $r \in [0, 1.2]$ is a random number drawn from a uniform distribution, and P_1 and P_2 are the two parents respectively. The intuition of line crossover is that the linear interpolation or extrapolation of the two parents may provide competitive offsprings [32]. In our case, we expect the line crossover is functionally similar to shape morphing which blends the geometric representation of two models together. The weight r tells the similarity between the child representation and the parent P_1. Sample results of the reconstruction models from the latent feature interpolation and extrapolation are shown in Fig. 4. Note that the voxelized representation is used to show the coarse, binary versions of the designs that are generated from the latent space. The actual output of the network is an SDF.

Copyright © 2019 by ASME

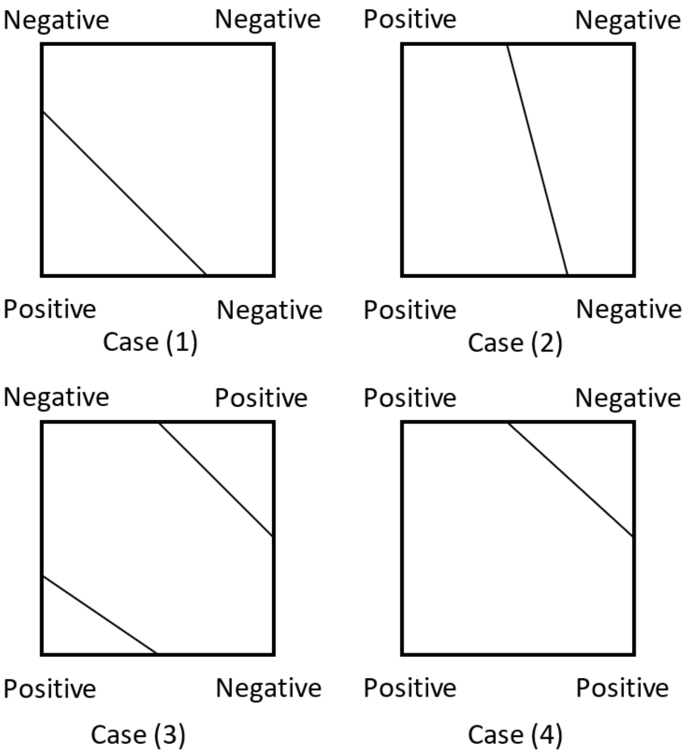

FIGURE 3. Four cases of arc generation in a rectangular face of a lattice cell.

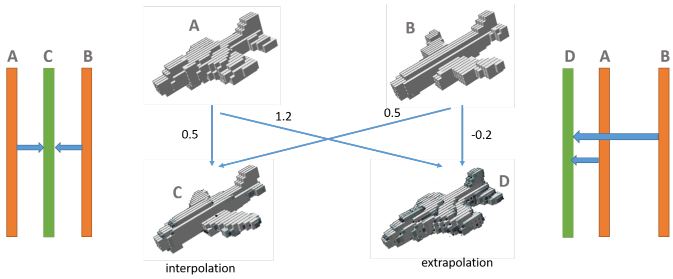

FIGURE 4. Sample reconstructions of the latent vector interpolation and extrapolation.

We first compute the scores for all the models in our dataset. Then, we rank all models and randomly select one model from every (N/n) interval where N is the number of total models, n is the population size (set to 100 in our approach). The probability of crossover and mutation is 0.9 and 0.05 respectively. The optimization goal is to minimize the Mean Square Error:

$$MSE = ||y_t - y_c||_2, \tag{4}$$

where y_t denotes the target functional objective score and y_c denotes the scores of the children generated in each generation.

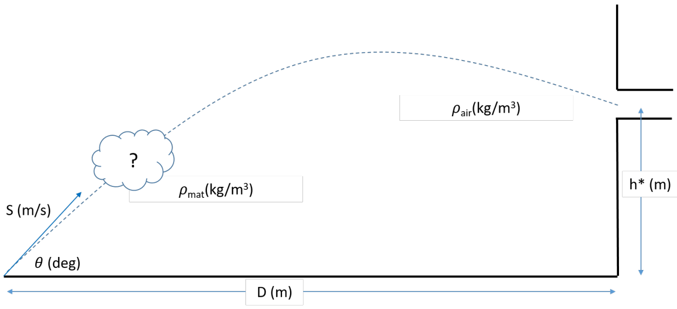

FIGURE 5. A schematic graph of the design task.

CASE STUDY

In this section, we demonstrate our approach on glider aircraft design. All the modules in the pipeline are specified from the suggested options we mentioned above. VSL, YS FLIGHT SIMULATOR and Genetic Optimizer are chosen as the three main modules in our pipeline.

For the data, we use the 4096 airplane models from the ModelNet40 repository [2]. We clean the floating patches in the models and align all the models in the same direction. Then we compute the SDF from the polygonal model to produce the network input $[41 \times 41 \times 41]$ in size for each model. It takes 3 hours to train the network for 1000 epochs on an NVIDIA GeForce 1080 GPU. The learning rate is 5×10^{-3}, and the batch size is 64.

Design Task

As shown in Fig. 5, given the initial launch speed $S(m/s)$, the initial pitch angle $\theta(degree)$, the density of the projectile aircraft $\rho_{mat}(kg/m^3)$ and the density of the air $\rho_{air}(kg/m^3)$ are given, the goal is to design the shape of the glider so that it can go through the gap located at h^*m high from the ground, and Dm horizontally away from the launch point.

In this design task, the constraint is that the projectile candidates should all fit in a $1m \times 1m \times 1m$ box without any propulsion. Among all the parameters, D, ρ_{mat} and ρ_{air} are fixed. S, θ as well as h^* can be set arbitrarily by the user. Designers are usually accustomed to the inverse task: Given the projectile, adjust S and θ to hit the target height, which can be regarded as a tuning process of several parametric design variables. By contrast, in our design problem, the shape of the projectile aircraft cannot be readily parameterized or represented with a limited number of design variables.

RESULTS

In our experiments, we set $\theta = 10^o$, $S = 45.7m/s$, $\rho_{mat} = 1000kg/m^3$, $\rho_{air} = 1.29kg/m^3$ and $D = 100m$. Our main interests lie in the height of the gap $h^*(m)$. We varies the value of h^* to test the performance of our proposed pipeline.

Copyright © 2019 by ASME

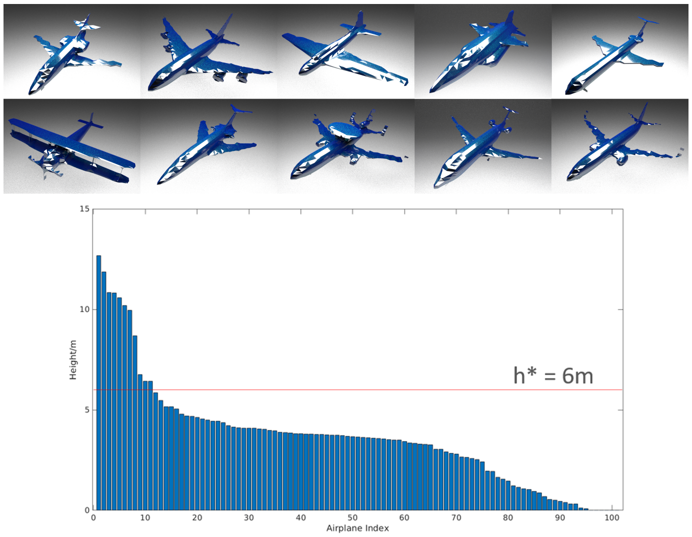

FIGURE 6. **Upper:** 10 randomly selected aircraft SDF models from the initial population. **Lower:** The landing height $h(m)$ distribution of the 100 initial population.

To have an intuitive sense of a reasonable range for $h*$, we randomly select 100 aircraft models from the dataset. Then we obtain the landing height of each aircraft model using the physics-based simulator we introduce in the CASE STUDY section. The acquired height range is $h \in [0, 12.6]m$. A plot of the sorted height distribution is shown in Fig. 6. Note that zero means the model lands at the ground before $D = 100m$.

First, we set $h^* = 6m$, which is around the mean value of the height range. After about 200 rounds of an iterative optimization process, the distribution of the landing heights for the final design candidates are demonstrated in Fig. 7. Unlike the diverse distribution in Fig. 6, the distribution is much flatter and mostly gather near the target height of $h^* = 6m$. In fact, if the tolerance is 0.1 m, 76% of design candidates in the final population satisfy the design requirement. (See Tab. 1) In the initial population, only 1% of the population can fulfill the same requirements. Fig. 7 also shows the diversity in the final synthetic populations, which will benefit the human designers with more valid options to choose from.

To demonstrate the ability of our pipeline in design space exploration, we intentionally set a design requirement that is originally unfeasible in the existing design candidates. Specifically, we set $h^* = 13.8m$. The height distribution of the suggested design candidates after optimization is shown in Fig. 8. We can observe that all the heights are within $[13.2, 13.9]m$, which exceed the maximum height($12.6m$) the initial designs can reach. According to Tab. 2, although the concentration of heights for final designs is not as impressive as $h^* = 6m$ case, 88% of candidates are able to fulfill the requirement if the tolerance is $0.5m$.

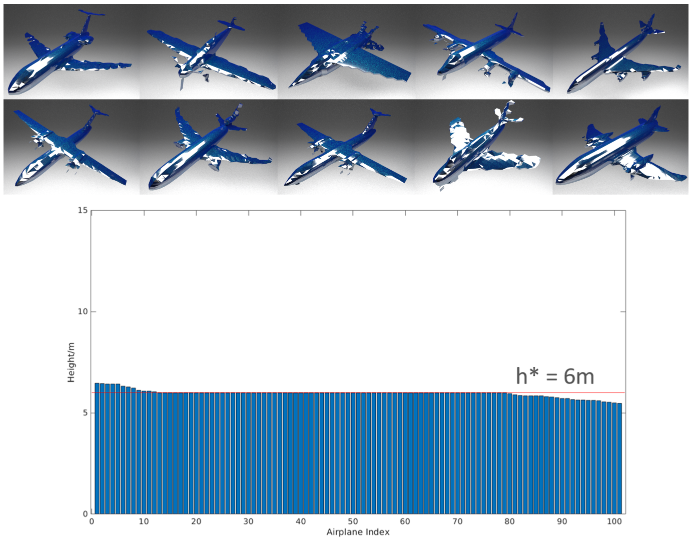

FIGURE 7. **Upper:** 10 randomly selected aircraft SDF models from the optimized population when $h^* = 6m$. **Lower:** The landing height $h(m)$ distribution of the 100 optimized population when $h^* = 6m$.

TABLE 1. PERCENTAGE OF THE DESIGN CANDIDATES WITH THE LANDING HEIGHT h, S.T. $|h - h^*| < \delta$ WHEN $h^* = 6m$.

$\delta(m)$	Initial	Final
0.1	1%	76%
0.5	4%	98%

The results also reveal the difficulty of exploration outside the original feasible space. To make the task even more challenging, we set $h^* = 14.5m$ and repeat all the design process. Expectedly, see Fig. 9, the final synthetic candidates cannot reach such a height requirement. But we obtain a design boundary reference in a data-driven manner as well as plentiful promising designs that provide useful suggestions for further manual design exploration.

CONCLUSIONS

This work presents an integrated conceptual design pipeline involving a data-driven shape learner, a function evaluator and a functional design optimizer. The pipeline is verified with a case study on the shape design of projectile aircraft models. When the design objective is set within the objective range of the design exemplars, our algorithm is able to synthesize a large set of design candidates that also satisfy the functional design requirements. It

Copyright © 2019 by ASME

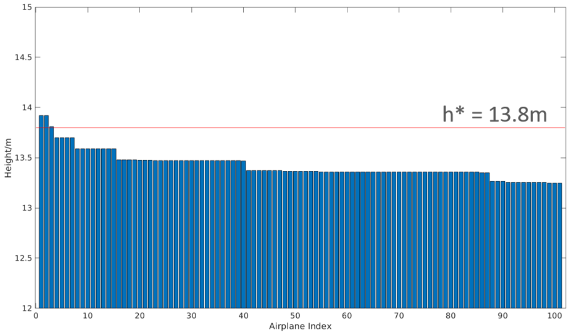

FIGURE 8. The landing height $h(m)$ distribution of the 100 optimized population when $h^* = 13.8m$.

TABLE 2. PERCENTAGE OF THE DESIGN CANDIDATES WITH THE LANDING HEIGHT h, S.T. $|h - h^*| < \delta$ WHEN $h^* = 13.8m$.

$\delta(m)$	Initial	Final
0.1	0%	5%
0.5	0%	88%

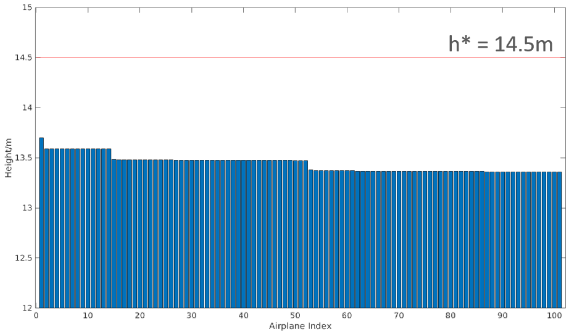

FIGURE 9. The landing height $h(m)$ distribution of the 100 optimized population when $h^* = 14.5m$.

is capable of generating new designs whose performance scores are outside the range of the original models.

Likewise, the same approach can be utilized to generate valid design candidates in other domains when adopting different options for each module. For example, if the design focus is the shape of a 2D beam bridge with load constraints, the shape learner alternative can be a 2D VAE. The physics-based simulator should accordingly be an FEA simulator. The functional design optimizer can adopt PSO.

At the meantime, even for design problems with the same dimension (2D/3D), various data representations can be introduced in the pipeline in terms of the focus area. Depth images are suitable when designing shell structures. While point clouds are beneficial when designing adjacent mesh surfaces. These potential extensions of our pipeline are becoming increasingly promising as several recent learning frameworks like Matterport3D [34] and PointNet [35] have been developed.

LIMITATIONS AND FUTURE WORK

Two time-consuming processes in our pipeline are the training of the shape learner and the geometry-based or physics-based simulation. A potential more efficient method is to directly establish a mapping between the latent feature vector and its corresponding physic properties, which may be a trained network as well. In this manner, designers can completely get rid of the original data format (2D/3D) during the iterative optimization process. The final proposed designs will be reconstructed once the objective is fulfilled.

When manipulating the feature vectors in the functional optimizer module, we don't have a sense of the exact meaning of each dimension in the latent space. This is a common problem which has also problematic for researchers from the machine learning field. Recently, researchers like Wieczorek et al. [36] are exploring learning orthogonal latent features, which may enable the designers to acquire a desired latent space.

REFERENCES

[1] Li, Y., Pirk, S., Su, H., Qi, C. R., and Guibas, L. J., 2016. "FPNN: field probing neural networks for 3d data". In Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, pp. 307–315.

[2] 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 Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1912–1920.

[3] Sedaghat, N., Zolfaghari, M., Amiri, E., and Brox, T., 2017. "Orientation-boosted voxel nets for 3d object recognition". In British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017.

[4] Brock, A., Lim, T., Ritchie, J. M., and Weston, N., 2016. "Generative and discriminative voxel modeling with convolutional neural networks". *CoRR,* ***abs/1608.04236***.

[5] Stutz, D., and Geiger, A., 2018. "Learning 3d shape completion from laser scan data with weak supervision". In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pp. 1955–1964.

[6] Sosnovik, I., and Oseledets, I. V., 2017. "Neural networks for topology optimization". *CoRR,* ***abs/1709.09578***.

[7] Guo, T., Herber, D., and Allison, J., 2018. "Reducing evaluation cost for circuit synthesis using active learning".

[8] Raissi, M., Perdikaris, P., and Karniadakis, G., 2018. "Multistep neural networks for data-driven discovery of nonlinear dynamical systems".

[9] Umetani, N., Koyama, Y., Schmidt, R., and Igarashi, T., 2014. "Pteromys: Interactive design and optimization of free-formed free-flight model airplanes". *ACM Trans. Graph.,* ***33***(4), July, pp. 65:1–65:10.

[10] Mark Fuge, B. P., and Agogino, A., 2014. "Machine learning algorithms for recommending design methods". *Journal of Mechanical Design,* ***136***(10), Aug.

[11] Chen, W., and Fuge, M., 2018. "Béziergan: Automatic generation of smooth curves from interpretable low-dimensional parameters". *CoRR,* ***abs/1808.08871***.

[12] Chen, Y., Tu, S., Yi, Y., and Xu, L., 2017. "Sketch-pix2seq: a model to generate sketches of multiple categories". *CoRR,* ***abs/1709.04121***.

[13] Dering, M. L., and Tucker, C. S., 2017. "A convolutional neural network model for predicting a product's function, given its form". *Journal of Mechanical Design,* ***139***(11), p. 111408.

[14] Burnap, A., and Hauser, J., 2018. "Predicting "Design Gaps" in the Market: Deep Consumer Choice Models under Probabilistic Design Constraints". *arXiv e-prints*, Dec, p. arXiv:1812.11067.

[15] Raina, A., McComb, C., and Cagan, J., 2018. "Design strategy transfer in cognitively-inspired agents". In ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. V02AT03A018–V02AT03A018.

[16] Burnap, A., 2016. "Estimating and exploring the product form design space using deep generative models". In DAC 2016.

[17] Oh, S., Jung, Y., Lee, I., and Kang, N., 2018. "Design automation by integrating generative adversarial networks and topology optimization". In ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. V02AT03A008–V02AT03A008.

[18] Viswanath, A., J. Forrester, A., and Keane, A., 2011. "Dimension reduction for aerodynamic design optimization". *AIAA journal,* ***49***(6), pp. 1256–1266.

[19] Viswanath, A., Forrester, A., and Keane, A., 2014. "Constrained design optimization using generative topographic mapping". *AIAA journal,* ***52***(5), pp. 1010–1023.

[20] Cinquegrana, D., and Iuliano, E., 2018. "Investigation of adaptive design variables bounds in dimensionality reduction for aerodynamic shape optimization". *Computers & Fluids,* ***174***, pp. 89–109.

[21] Chen, W., Chiu, K., and Fuge, M., 2019. "Aerodynamic design optimization and shape exploration using generative adversarial networks". In AIAA SciTech Forum, AIAA.

[22] Liu, S., Giles, C. L., and Ororbia, A., 2018. "Learning a hierarchical latent-variable model of 3d shapes". In 2018 International Conference on 3D Vision, 3DV 2018, Verona, Italy, September 5-8, 2018, pp. 542–551.

[23] Edwards, H., and Storkey, A., 2016. "Towards a neural statistician". *arXiv preprint arXiv:1606.02185*.

[24] Maturana, D., and Scherer, S., 2015. "Voxnet: A 3d convolutional neural network for real-time object recognition". In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 922–928.

[25] Kingma, D. P., and Welling, M., 2013. "Auto-encoding variational bayes". *arXiv preprint arXiv:1312.6114*.

[26] Nair, V., and Hinton, G. E., 2010. "Rectified linear units improve restricted boltzmann machines". In Proceedings of the 27th international conference on machine learning (ICML-10), pp. 807–814.

[27] Ioffe, S., and Szegedy, C., 2015. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, pp. 448–456.

[28] Mitchell, T. M., 1997. *Machine Learning*, 1 ed. McGraw-Hill, Inc., New York, NY, USA.

Copyright © 2019 by ASME

[29] Yamakawa, S., 2019. Ys flight simulator. `http://www.ysflight.com`. The program is available online since Feburary 2019.

[30] GERO, J. S., and KAZAKOV, V., 2000. "Adaptive enlargement of state spaces in evolutionary designing". *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* ***14***(1), p. 3138.

[31] Cvetkovic, D., and Parmee, I., 1999. "Genetic algorithms based systems for conceptual engineering design".

[32] Rasheed, K., Hirsh, H., and Gelsey, A., 1997. "A genetic algorithm for continuous design space search". *Artificial Intelligence in Engineering,* ***11***(3), pp. 295 – 305.

[33] Wright, A., 1999. "Genetic algorithms for real parameter optimization". *Foundations of Genetic Algorithms,* ***1***, 06.

[34] Chang, A. X., Dai, A., Funkhouser, T. A., Halber, M., Nießner, M., Savva, M., Song, S., Zeng, A., and Zhang, Y., 2017. "Matterport3d: Learning from RGB-D data in indoor environments". In 2017 International Conference on 3D Vision, 3DV 2017, Qingdao, China, October 10-12, 2017, pp. 667–676.

[35] Qi, C. R., Su, H., Mo, K., and Guibas, L. J., 2017. "Pointnet: Deep learning on point sets for 3d classification and segmentation". In 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pp. 77–85.

[36] Wieczorek, A., Wieser, M., Murezzan, D., and Roth, V., 2018. "Learning sparse latent representations with the deep copula information bottleneck". *CoRR,* ***abs/1804.06216***.

Copyright © 2019 by ASME