中文摘要

英文摘要

第一章 仿生结构设计

第二章 生成对抗网络

2.2生成对抗网络

2.3 dcgan

2.3 wgan-gp

2.4 igan

第三章 实验

3.1 数据集构成与增强

3.2 训练过程

3.3 限制给出过程

3.4 实验结果，对比

第四章 结论

4.1结论

4.2局限性

4.3 发展方向

4.4 总结

中文摘要

仿生结构设计旨在利用生物的结构特征指导工业工程中的产品设计，以达到节约原材料，优化力学性能等目的。3D打印技术的发展大大拓展了结构设计的空间，但从物种中抽象出仿生结构的过程仍然缺乏普适的理论指导。现有的抽象方法主要依赖于相关专家的主观意见，不同方面的仿生需要不同的专业知识，人力成本过高。且主观性太强，不能保证充分利用生物的进化成果。本文提出了一种新的抽象仿生结构的方法，利用生成对抗网络学习生物的结构特征，再基于人为限制产生符合要求的仿生结构。

英文摘要

第一章

1.1仿生

大约46亿年前，地球从一片混沌之中诞生了。天地之初，一片死寂，但在漫长的演化过程之中，在无数巧合的重重叠加之下，原始的生命出现了。此后，又经历了数十亿年的自然选择，优胜劣汰，动物和植物不断地适应着周遭的环境，并不断改造着环境。最终，生命和环境达到了巧妙的平衡，并不断的进化了下去。在这一进化过程之中，为了在残酷的竞争中胜出，动物和植物不得不改进自身的结构特征，有的进化出具有迷惑性质的外形逃脱天敌的追捕；有的进化出坚硬的外壳承受外部压力；有的进化出密集的孔隙结构来减少资源的消耗；有的进化出复杂的脉络增加运输效率……我们身处的大自然正是地球46亿年来的智慧结晶。

相比之下，人类的进化史不过百万年，人类文明的诞生仅仅过了大约一万年，和自然相比可以说是微不足道。自然界正是我们的老师，是人类各种科学技术，工程艺术的灵感来源。人类的仿生可以追溯到十多个世纪之前，如锯子的发明被认为是受到了叶片锯齿形边缘的启发。人类文明的发展中伴随着无数的仿生。现代仿生学的诞生可以追溯到1960年，美国的J.E.Steele提出Bionics一词。仿生学研究生物系统的结构和特性，生物的行为，生物和环境的相互作用等，为工业工程设计提供全新的设计思路，系统原理的一门学科。

仿生学包括很多方向，主要有形态仿生：模仿生物的外形，色彩等；结构仿生：模仿生物的力学结构；材料仿生：模仿生物的微观结构设计材料。仿生学已经成为数学，生物学，工程科学，材料化学等多学科相互交叉，相互渗透的重要新兴学科，有着广阔的发展前景。

1.2 生物微结构与轻量化设计

工程中经常需要在重量，材料消耗量和性能表现中寻求平衡，因此需要人们不断设计并优化力学结构，这催生出轻量化设计（lightweight design）的需求. 生物在进化的过程中倾向于设计出能最大程度地节省能量消耗和材料用量，同时还保持优秀力学性能的结构，这与轻量化设计的理念不谋而合。大量文献从生物的结构特征中获取灵感，设计出了具有优秀力学性能，如抗压，抗弯，抗扭转特性的力学结构，广泛应用到了各种产品中。常见的仿生对象有：

1.2 Biological Micro structure and Lightweight Design

Engineering design usually need to reach a balance between weight, material consumption and its mechanical performance. Thus engineers have to optimize the topological and mechanical structure of their products. These requirements lead to the demand for lightweight design. During the evolution, creatures tend to create biological structures that can economize materials and energy, without hurting their mechanical performance. These phenomena match the idea of lightweight design perfectly. Biomimicking from nature may offer the potential for lightweight design. A wide range of creatures have been mimicked:

1.Timbers：Timbers hold excellent solidity and stiffness per unit mass, which results from timbers’ multilayer composite structure. Spiraling biological fibers, honeycomb or foam-like cellular cores, as well as other cellular structures form porous micro structures in timbers. As a result, the biological materials are saved, stiffness and solidity are enhanced. Timbers serve as important inspiration for lightweight design.

单位质量木材具有优秀的刚度和韧性。这源于木材的多层级复合结构。纤维丝的螺旋缠绕，木细胞的紧密排列使得微观上的木材具有多孔结构。这一结构不仅大大减少材料用量，增强了强度和刚度，还使得木材的性能有了各向异性。木材是轻量高强度结构的重要灵感来源。

2. Bones: Bones are critical biological parts that support creatures. Basically, they should possess good properties such as high load-carrying efficiency and less mass. Their form, shape and size have been optimized to meet these demands during evolution. Collagen proteins and mineralized wafers are cross-linked in bones, resulting in multilayer fissure structures. These fissures can conduct strains and dissipate energy efficiently. Also, anisotropy is produced to meet different mechanical requirements at different parts of creature.

2.骨骼：骨骼是生物重要的支撑组件，因此需要兼备轻量，坚韧等特点。骨骼中的胶原纤维和矿化晶片相互交联，形成多级的裂纹结构。这些裂纹能有效地传导应力，耗散能量，也使得骨骼的力学性能有了各向异性，满足生物不同部位的不同受力需求。

3. Carapaces: Carapaces of crabs and lobsters are another set of biological structures widely studied. Their carapaces possess excellent ability to prevent creatures from intense pressure and collision. This ability results from the multilayer structure and three-dimensional porous structure in the micrometer scale. Carapaces inspire many studies in designing lamination materials.

3.甲壳：螃蟹等动物的甲壳拥有优秀的抗撞击能力和抗压能力。这种能力源于甲壳的多层级结构和微米级纤维形成的三维孔洞结构，这为人们设计叠层材料提供了思路。

4. Plankton: Plankton such as diatom and radiolarian need to defense themselves form predators’ attack. Secondly, in order to receive sufficient sun light, they have to float in shallow water. The adversarial optimization has designed hulls and exoskeleton with very light weight without affecting their outstanding mechanical.

4.浮游生物：硅藻，放射虫等浮游生物的外壳和外骨骼正是力学性能优异的轻量化结构。这是因为：1.它们需要抵御多种捕食者的威胁2.为了接收足够多的阳光，它们需要漂浮在浅层水域中。重量和性能的相互妥协与优化产生出多种多样的几何结构。

从上面的综述可以发现，自然界生物的微结构主要有以下几种特点：

Several main features can be summarized from the biological structures above:

1. The raw materials used in creature’s biological structures are simple. Most of the biological structures achieve outstanding mechanical performance with single materials. Their delicately designed topological structures are the key of lightweight design.

2. The micro structures in creatures are usually porous and cribrate. These structures significantly reduce mass as well as guarantee mechanical solidity and stiffness. Without changing raw materials, creatures thus meet the demands of lightweight design.

3. Micro structures with diverse distributions correspond to different mechanical properties such as load-carrying efficiency, bending resistance ability and compression resistance, and in turn play a role in different situation. Micro structures have huge potential in lightweight design.

1.生物结构的原材料种类并不复杂，多数结构使用仅单一的材料就能拥有良好的性能。2.生物的结构在微观的尺度下都呈现出多孔结构。这种结构能够有效地降低密度。3.生物通过优化孔隙的分布尽可能地提高特定用途力学性能。

1.3 Additive Manufacturing

Additive Manufacturing is a fast manufacturing technique. It use metal powders and plastics as raw material and manufacture products by printing materials layer by layer. This technique create 3D objects directly from 3D CAD model with little restrictions about the shape of the object. This freedom in product design can be used widely to significantly enhance the functionality of series products by substituting traditional parts with additively manufactured parts. As a result, people are no longer restricted to structures with simplest topology. Products with complex structures become reliable.

3D打印技术是一种快速成型技术，它利用粉末金属和塑料等打印材料通过逐层打印的方式制造出软件设计的模型。3D打印技术可以打印的模型范围十分广泛，基本能够将各种设计需求加以实现。这大大地拓展了工业中产品结构设计的思路，人们不必局限于简单的结构，可以自由地设计并实现需要的力学结构。

Additive Manufacturing’s geometrical freedom of design makes structures even as complex as those in creatures possible to produce. Moreover, contrast to conventional manufacturing, Additive Manufacturing serves as an ideal technology for rapid prototyping and rapid manufacturing. All of these benefits makes AM a perfect method in structural bionic design, and have been applied to biomimic varied structures.

由于其优异的性能，3D打印技术被广泛地运用到结构仿生和轻量化设计中。它不仅能让从前难以制造的各种生物结构，特别是多孔微结构的制造成为可能，而且能够快速得到样品。这为力学结构的分析和性能优化提供了极大的便利。许多文献在结合3D打印技术与结构仿生方面进行了探索，将结构仿生技术提升到了新的层次：

1.4 Current Structural Bionic Design

现有的结构仿生论文虽然数量众多，但是它们的主要流程大致可以概括为3个步骤。1.寻找合适的物种，通过各种技术手段记录其结构特征。2.对这些结构进行细致的力学分析，总结其性能优异的原因。尝试对生物的结构进行抽象化，设计出暂定的仿生结构。 3.制造产品，利用有限元分析等手段进行力学性能测试，根据反馈进行拓扑优化，重新调整仿生结构，直到达到性能要求为止。

Although a lot of papers have explored structural bionic design with Additive Manufacturing, the abstraction of complex biological lightweight structure into a producible component is still a significant and fundamental step in the transfer of design principles from nature to technique lightweight solutions. Currently the procedures of structural bionic design commonly have three steps:

Step 1. Look for suitable creatures to mimic with respect to mechanical goals and collect samples.

Step 2. Analyze biological structure with mechanical and biological knowledge, then abstract the structures and design products.

Step 3. Manufacture the products, and test their mechanical performance. According to the feedback, make topological optimization or redesign the products until meeting requirements.

The major obstacles of currently procedure of abstraction are the particular geometry of various biological structure. Actually, the abstraction of biological lightweight structure remains to be an arbitrary component. That is to say, the abstraction results critically depend on the expert who execute the abstraction. The disadvantages of this method are obvious:

Firstly, it needs sufficient expertise to execute the abstraction. The person should master enough biological knowledge to understand the biological structure of creatures. Also, he should master enough knowledge of structural mechanics to analysis what structures lead to biological structure’s outstanding mechanical performance. Moreover, transforming biological structure into mechanical structure while remaining its advantages needs skillful engineering experience. These requirements make a structural bionic design task cost highly on experts in relative fields.

Secondly, the transfer capability of structural bionic design is weak. Nature has extremely diverse Biological structures. However, the experience of biomimicking on one structure cannot be transferred directly to other structures. What’s more, for different mechanical purposes and applications, the abstraction should follow different guidelines. People have to repeat the procedure to execute abstraction.

Thirdly, although people even can invite the best expert to execute the abstraction, still he might fail to make full use of biological structures’ mechanical potential. Human beings benefit from knowledge and experience, but also are restricted by them.

Fourthly, micro biological structures like porous structures have latent but important connection to their shape and exterior. However, human experts are hard to capture this relationship. So in order to guarantee mechanical performance, the shape of structural bionic design are usually fixed, bringing constrains on product design. How to adjust the shape of abstraction naturally, i.e. without affecting the coordination between exterior and interior’s micro structures remain unsolved.

流程图

可以看出，这种设计流程的核心步骤是第2步中的分析和人为抽象化，抽象化的质量直接决定了产品的力学性能。但这一流程存在很多缺点：1. 人力成本过高：力学分析和人为抽象过程需要相当专业的知识，要能准确地分析出怎样的结构能够更好地继承生物的特征。 2.迁移性太差：生物结构多种多样，不同结构，不同用途的仿生需要的专业知识也不尽相同，需要对该结构进行独立的分析。其他仿生结构设计的经验无法直接用于指导新的仿生。3.仿生效果受限：人为抽象的主观性太强，仿生效果不稳定。且人们可能受到经验和知识的束缚，不能保证充分发挥生物结构的潜力。4.设计自由度低：由于生物体的特定力学结构，特别是微观孔隙结构等往往和生物结构外部形状绑定，但生物体的外形不一定是产品所需要的外形。怎样自然地对仿生结构的外形和轮廓进行修改，同时避免破坏内部多孔结构的协调性，避免性能的损失是人为设计很难做好的。

In order to solve all the obstacles above, in this paper we propose a brand new design method that satisfies properties below:

1. Universality: we propose a universal procedure to execute structural bionic design. There’s no need to transfer biomimicking experience anymore.

2. Non-artifact: Our algorithms learn the features from biological samples automatically, getting rid of the reliance on huge amount of expertise.

3. Natural generating: Our method enables users to give constraints such as shape on output products, and automatically generate natural samples that combine biological structures with the constraints.

4. Optimize during abstraction: Instead of making topological optimization after abstraction, we regard mechanical performance as constraints during abstraction.

为了解决上述问题，我们需要一种普适性的方法来指导仿生结构设计，实现结构仿生的去人为化，达到节省人力成本，改进仿生效果，增加设计自由度的目的。这要求我们的算法能够。自动学习理解生物体的结构特征，并且能复现出这些结构 2.能够根据用户的输入（如颜色，形状等）生成出带有特定特征的仿生结构。3.生成的结构要能自然地结合生物特征和用户输入，减少力学性能的损失。

1.3 本文思路

In this paper, we propose a biomimicking procedure based on Generative Adversarial Networks (GAN).

本文提出利用深度学习中的生成对抗网络学习生物的结构特征，并将用户输入视作限制对生成结果进行优化，最终产生出符合用户需求的仿生结构。

2 Deep learning and Generative Adversarial Networks

2.1深度学习

Deep learning are machine learning techniques that have been generally applied in image recognition, speech recognition, language processing and etc. Deep learning design computational models with multiple layers to learn multiple representations of the natural data. The construction of traditional machine learning and pattern recognition system requires considerable expertise in relative domains to design a useful feature extractor to transform the raw data into feature vectors which the subsequent system such as classifier and clusterer could handle.

Thus the processing results highly rely on the design of feature extractor. On the other hand, deep learning methods compose many non-linear modules which each could transform a level of representation into a higher and more abstract level, and finally learn multiple layers of representation. Moreover, according to the performance of the representation, deep learning network could optimize the parameters of the non-linear modules. As a result, deep learning get rid of the dependence on considerable expertise. Since the extractors are not designed by human engineers, and are learned with raw data, we can apply deep learning methods to many aspects easily. More importantly, they might have far better performance on a specific task than before. In fact, while learning these transformations, deep learning network could approximate very complex functions, which are valuable in various applications.

深度学习是近年来广泛应用于图像识别，语音识别，自然语言处理等方面的机器学习技术。它通过训练多层的计算模型来学习原始数据的多层表示。传统的机器学习算法需要相关专家根据原始数据的特点设计出相应的特征提取器，将原始数据表示为特征向量，再通过分类，聚类等算法进一步处理。机器学习的效果与特征器的设计密切相关。而深度学习在每一层模型中集成了大量的非线性模块，将输入转化为新的表示，这些表示又作为输入被下一层非线性模型处理。层层的处理从原始数据中提取出简单的表示，又将这些简单的表示组合成更深层次的表示。根据数据处理的效果，又反过来优化这些非线性模块的参数，从而改进特征的提取方法。这么一来便实现了，特征提取器设计的去人为化和自动优化，大大提升了特征提取能力。深度学习的网络可以视作拟合能力非常惊人的非线性系统，通过这一系统的优化，有望突破人为设计算法的瓶颈。

2.2卷积神经网络

深度学习有多种不同的算法模型，但模型总体上可以分为三个层次：输入层，即输入的原始数据；隐藏层：数据的

2.2 Convolutional Neural Networks  
 Deep learning is a large category containing many sub-frameworks, such as Convolutional Neural Networks, Auto Encoding, Sparse Coding, Boltzmann Machine, Deep Believe Networks and Recurrent Neural Network. When dealing different tasks, we should choose suitable framework.

Convolutional Neural Networks(CNN) are powerful tools to process machine learning problems in fields like image recognition, especially when the size of image is large or we need to achieve high recognition rate.

A neuron network consist of many layers. Rather than regarding each layer as representing a vector-to-vector function, we can also regard it as a system consisting of many units that act in parallel. Each unit imitate a neuron in the sense that it takes many inputs and calculate its activation values based on its weights and biases. Since the parameters like weights and biases are trainable, we could update these vector-to-vector functions continuously. The ideas of using these layers of vector valued functions are loosely inspired by neuroscientific observations, while model neural network are guided by mathematical and engineering principles. Rather than regarding neural networks as modelling of human brains, it’s better to regard them as powerful nonlinear systems that works as function approximation machines designed to achieve statistical generalization.

A simple Convolutional Neural Networks contains three parts: input layer, hidden layer and output layer. The input layer represent raw data, and the output layer represent the results. The specific hidden layers used in Convolutional Neural Networks are convolutional layers. Convolution in deep learning generally refers to a discrete convolution option define on tensors. We can regard convolution as multiplication by a matrix. For example, If we have a two-dimensional image, the convolution with a two-dimensional kernel K is defined as：

Convolutional layers are designed with some important princilples:

Sparse interactions: Traditional neural networks use matrix multiplication to describe the interaction between inputs and outputs. Each input is connected to each output. For example, an image may contain millions of pixels. However, we actually only need tens of or hundreds of pixels to represent small but meaningful features such as edges and shapes. So we only need to use much fewer parameters to extract the features, which is going to save memory, reduce operations and enhance statistical efficiency. By doing convolution operations, we could construct only sparse interaction between units by making the kernel size smaller than the inputs.

Parameter sharing: A specific feature may appear at many location. Rather than using different sets of parameters for different location, we could fix kernels and use the same set of parameters for one feature. This idea not only greatly reduces the parameter number, but also makes training faster. Moreover, each kernel thus is corresponded to a specific feature, making it interpretable and invariant under location shifting.

Subsampling: In order to reduce parameters, people apply subsampling operations such as maximum polling and average pooling when designing convolutional neural networks. Pooling over spatial neighborhood can produce invariance to translation. When applied to discrete convolutions, pooling will help features to decide which translation to ignore.

Multiple convolution layers: In practice, there are many convolution layers. Units in deeper convolution layer can interact with (though indirectly) a larger amount of inputs. Thus depth in networks produce invariance and robustness. Since each convolution layer is often followed by a non-linear translation, deeper networks have better performance in fitting complex functions.

2.3 Generative Adversarial Networks

With the belief to understand our complex world in a probabilistic way, deep learning methods aim at discovering hierarchical models that can represent the probability distribution of natural data such as natural images and audio waveforms. Deep learning consist of discriminative methods and generative methods. Discriminative models usually learn to map raw, sensory and high-dimensional data into meaningful class labels, and have got huge success in many classification tasks. Generative models learn how to use inputs with distributions we have known to produce distributions as similar as possible to the data. The inputs are often low-dimensional and can be described with much smaller set of parameters. Generative methods force models to discover and abstract the hidden nature of data and then reproduce the data. Common generative methods includes restricted Boltzmann machines (RBMs), deep Boltzmann machines (DBMs), Deep belief networks (DBNs), Variational Autoencoder (VAE), and Generative Adversarial Networks (GAN).

2.3.1 Generative Adversarial Networks

As is shown in the graph, a general GAN framework consist of two parts, Generator and Discriminator. Generator and Discriminator are both multilayer perceptions. The inputs of the Generator are random noises z of a prior distribution, say uniform distribution or normal distribution. Then the Generator maps the inputs to data space with differentiable neural networks parameterized by theta\_g.

The Discriminator takes inputs x from data space, and outputs a scalar D(x) representing the probability that x comes from the data rather than the outputs of Generator. The task of Generator is to generate as similar as possible distributions of the real data to deceive Discriminator. That is making D(G(z)) give the same scalar as D(x). While the Discriminator aims to separate real data from data space and fake data from Generator, and maximize the probability to assign correct labels .

To be specific, we train Generator to minimize log(1-D(G(z))), and train Discriminator to maximize log(D(x)). In other words, D and G play a min-max game with value function V(G,D):

We could solve this min-max game by updating parameters of G and D alternately with stochastic gradient descending.

(1) Proposition 1.

(2) Theorem 1

(3)Proposition 2

Graph () shows us how GAN framework works: The black, dotted line represents samples from the data generating distribution p\_data. The green, solid line represent the generative distribution P\_g(D) produced by generator. The blue , dotted line represents the discriminative distribution which discriminates between real and fake samples. Generative Adversarial Networks are train by simultaneously updating G and D. IF D and G have enough capacity, the generative distribution p\_g finally converges to p\_data, and Discriminator won’t be able to differentiate the two distributions, making D(X)==1.

Here are some performance of GAN on Mnist dataset.

2.3.2 Deep Convolutional Generative Adversarial Networks

There is no need to define a heuristic loss function in GAN framework, making its training extremely unrestrained. However, it also makes traditional GAN model hard to converge. The training process of GAN is unstable, the Generator tends to produce meaningless outputs, so Discriminator can easily separate real samples and fake samples, resulting in useless feedback to Generator, making the latter even perform worse.

In order to guide the training of GAN, XXX and XXX propose a set of constrains from convolutional networks on the architecture of GAN. They call these architectures Deep Convolutional GANs, and have shown that DCGANs are strong candidates for unsupervised learning on image datasets.

The architecture of DCGAN is shown below:

The Generator takes a 100-dimensional random uniform distribution noise as inputs and then map them into a small spiral convolutional representation with a large number of feature maps. After a series of fractionally-strided convolutions, the inputs are rescaled into a 3-channel 64\*64 pixel image.

Since DCGAN utilizes CNN architecture, the training stability is enhanced. Also, while training, the Discriminator learns meaningful representations with its convolution layers. For example, DCGAN can capture elements like bed, windows in LSUN bedrooms dataset.

Moreover, DCGAN has shown that the manifold learned has smooth transitions. We could walk in the latent space to adjust the outputs of Generator. For example, the author have shown that interpolation of random points in Z produces meaningful image morphing. They also apply vector arithmetic to integrate different features.

2.3.3 Wasserstein GAN

Traditional GAN framework is well known to be unstable. DCGAN attempts to solve this problem by restricting the network structure with CNN, while WGAN theoretically analyzes the reason of traditional GAN’s delicate training process.

The author points out that, when the Discriminator reaches its optimum, optimizing the loss functions of GAN is equivalent to optimize distance functions related to the Kullback-Leiber divergence KL(P\_r||p\_thata) and Jensen-Shannon divergence JS(P\_r||P\_theta). However, when the model manifold and the true distribution’s support have a non-negligible intersection, KL divergence is infinity, while JS divergence is log2. This leads to gradient vanishing problem when updating Generator. Since the manifold of Generator outputs’ support are usually low-dimensional submanifold of a high-dimensional space, the measure of the intersection of P\_r and P\_g tends to be very small. So gradient vanishing is an inevitable problem in origin GAN, producing imbalance in Generator’s and Discriminator’s training process. What’s more, the infinity of KL divergence will force the Generator to produce real but repetitive samples, resulting in mode collapse problem.

In order to solve the disadvantages of origin GAN, the author comes up with Earth-Mover distance

Where .. denotes the set of all joint distribution ... whose marginals are .. and .. . Intuitively, EM distance indicates how much ‘mass’ must be transported form x to y to transform distribution P\_r to P\_g. In fact, EM distance is the cost of optimal transport plan theory.

A simple example shows the advantages of EM algorithm: Let Z ~ U[0,1] be a uniform distribution on [0,1],and set P\_0 = (0,Z), P\_theta=(theta,Z) be distributions in R^2, where theta is a fixed real parameter. Then:

This example show us sequences of distributions that doesn’t converges under other distance will converge in EM distance. When learning these kinds of distributions over a low-dimensional manifold using gradient descending, other distances cannot give sufficient gradient information since the supports tend to have non-empty intersection contained in a set of measure zero. But we could still use EM distance to train the model.

From EM distance to WGAN : Since the joint distribution in EM distance cannot be solved directly, they use a lemma to get an equivalent distance function.

Where the supreme is taken over all the K-Lipschitz functions. Since deep neural networks have incredible ability to approximate functions, we could use a network to represent f and expect the result will be sufficiently close to EM-distance. Origin WGANs use weight clipping to enforce Lipschitz constraints, and Improved WGANs use gradient penalty to enforce them.

Since the Discriminator is trained to give EM-distance of P\_ra and P\_g, its loss function d\_loss can give a meaningful metric to indicate training stage. Also, the good properties of EM-distance enable WGANs to have a flexible choice of network structure without apparently affecting its performance.

2.3.4 iGAN

3 Approach and Model Architecture

3.1 Prior Steps

In order to execute structural bionic design, we first need to obtain sufficient data about the chosen biological structure. Commonly used descriptions of biological structures include cross-section photos taken with camera (timbers), CT photos (bones), and etc. Surely, these photos should be taken skillful to eliminate irrelevant factors. If necessary, we also could carry out pre and post processing.

In a word, after a skillful collecting and processing, finally we can get enough natural image data about our target biological structure.

3.1 Learning the Natural Manifold with GAN

If the image dataset has enough samples, we could expect it to fully reflect the biological structure. Let we assume that all biological structure images in our dataset are sampled from a distribution that lies on an ideal manifold M\_, with distance function that measures two images’ sensual similarity. Directly modeling this manifold is extremely difficult, since it is a complex-structured manifold in a million-dimensional space.

Notice that by training GAN on a dataset with latent input of relatively low dimension (say 100), we can approximate M\_ by a low-dimensional manifold M\_\_. The latent input z then serves as a set of global coordinates of M\_\_. We can formally define M\_\_ = {} and regard it as an approximation of M\_. Also, the distance function can be approximated by the Euclidean distance between latent vectors, i.e.,

GAN have been chosen to execute approximation because: 1. GANs, especially its variants such as DCGAN, WGAN have been observed to be able to generate high quality but brand new images similar to training set, so the manifold can be nicely approximated. 2. GAN network structures are differentiable, thus the Euclidean distance of latent vectors can be regard as distance functions.

3.2 Manipulating the Latent Vector

By now, we already have an approximate manifold M\_\_, and a global coordinate z. Then our goal is to manipulate the latent vector z to modify the image on M\_\_, so the generated image G(z) meets user’s requirements. We achieve this by updating z to match requirements.

3.2.1 Solving The Constraint Problem

Each of User’s requirements is formulated as a constaint f\_g(x)==v\_g; where g denotes a type of constraint, f\_g maps a sample x to a feature f\_g(x), and v\_g denotes the value of constraint user have set. And we define objective as follows:

The constraint term measures the deviation from the constraint. The smoothness term force choosing new images G(z) close to G(z\_0) on the manifold, since from a mathematical point of view, it’s hard to find a global coordinate description of manifold and instead we usually use a local chart. E\_D =… represent the visual realism of the generated output scored by the GAN Discriminator. This term forces the generated image close the M\_\_ and thus inherit biological structure.

Since the are manipulating the latent vector z to modify image on the approximate manifold M\_\_ to solve the constraint problem, the generated images are in fact chosen from M\_\_. So they naturally inherit the features of M\_\_, M\_ and in turn the raw biological dataset. This means the generated images combine biological features with human design naturally.

3.2.2 Constraints and Their Formulations

(1) Color: The dyeing of biological structure usually produce colors with gradual change based on the size and density of the micro structure. So colors can be used to guide and control the micro structure of products. For each pixel marked by user’s requirement, we set the mapping f to be f\_color(I) = I\_p, the rgb vector at the pixel and let the constraint to be f\_color(I) = I\_p=v\_p, forcing the selected pixel p to have value v\_p.\

(2) Shape: To meet diverse engineering demands, user would like to generate bionic structures in varied shapes. In our algorithm, shapes are also regarded as constraints. We use HOG feature which reflects gradient information to represent the shape of selected region. In this case, mapping f becomes a HOG feature extractor f\_hog, i.e f\_hog(I)=HOG(I)\_p, the Hog feature at location p. We set f\_hog(I) = HOG(I)\_p to be as close as possible to v\_g, the HOG feature of user’s stroke.

3.2.3 Gradient Decent Update

Since the structure of manifold are complex, for most constraints the equation(5) is non-convex. We use gradient decent update to solve it.

4 Experiments

4.1 Datasets

Since our bionic design method is pioneering, nobody has collected enough number of high quality images about biological micro structures. Useful datasets are very limited on the Internet. We have to apply data augmentation on limited images to prove the empirical validation of our methods.

4.1.1 Bamboo Cross-section Images

The primary image is a bamboo cross-section image shown in \_\_. We randomly crop 64\*64 patches on the primary image and apply rotation and flipping operations to generate new images with different angles. The resulting dataset consist of 24576 RGB images of size 64\*64. The patches are shown in \_\_.

4.1.2 Midrib Images

The primary images are three images of Symplocos mosenii from MidribDataset. We also randomly crop 64\*64 patches and apply rotation and flipping operations, but since we would like to focus on the specific biological structure of the edge of the midrib, we’ve wiped off a large portion of images that fail to contain the edge.

4.2 Details of Adversarial Training

4.2.1 Network Structure and GAN method

Since the dataset consist of images, we would like to include convolution layers to capture biological features of micro structures. We follow the same network design principle with DCGAN.

However, since our dataset are generated from several primary images, the quality of the dataset cannot match those used in DCGAN paper. In fact, directly applying DCGAN on our dataset may cause mode collapse, i.e. when varying latent vector z, the output remains unchanged.(see figure\_\_). Thus, we choose loss functions of WGAN, but still use DCGAN’s network structure.

In Generator, a 128 dimensional uniform distribution is projected to a convolutional representation with a large number of feature maps. Then a series of four fractionally-stride convolutions are executed to map these feature maps to a 64\*64 3-channel image. Every convolution is followed by a relu function as activation function to increase nonlinearity. In the end, we use a tanh function to scale the image into a color space between [-1,1]. Since we would like to use loss functions of WGAN, there is no need to add batch normalization.

In Discriminator, the inputs are scaled to [-1,1], and then a series of four convolutional layers map the inputs into feature representations and finally we get a scalar. In WGAN framework, we don’t need to use sigmoid function to transform this scalar into probability. Since the Discriminator serves as an approximation of EM-distance, the scalar represents an approximate distance between generative distribution and real sample distribution, indicating the realism of the image.

4.3 Solving User’s Constraints

4.3