Proceedings of the Seventh International Conference on
Artificial Intelligence, Soft Computing, Machine Learning and Optimization,
in Civil, Structural and Environmental Engineering
Edited by: P. Iványi, J. Kruis and B.H.V. Topping
Civil-Comp Conferences, Volume 11, Paper 4.2
Civil-Comp Press, Edinburgh, United Kingdom, 2025
ISSN: 2753-3239, doi: 10.4203/ccc.11.4.2

Machine Learning-Powered Geometry-Aware Filter: A Novel Human-Informed Approach for Advanced Topology Optimization

X. Zhuang[1], W. Zhang[1,2], X. Guo[1,2] and S.-K. Youn[3,1]

[1]State Key Laboratory of Structural Analysis for Industrial Equipment, Department of Engineering Mechanics, International Research Center for Computational Mechanics, Dalian University of Technology, China
[2]Ningbo Institute of Dalian University of Technology, Jiangbei District, Ningbo, China
[3]Department of Mechanical Engineering, KAIST, Daejeon, Republic of Korea

Abstract

This paper proposes a machine learning-powered Geometry Filter to enhance the design capabilities of SIMP-based topology optimization. By reconstructing the paradigm of traditional density filtering functions, the method establishes a filtering format with geometric feature modulation, enabling effective interaction between human design intent and the topology optimization process. To achieve this, machine learning is employed to construct multimodal geometric feature matching metrics, transforming explicit geometric elements, abstract stylistic features, and intuitive conceptual designs into mathematically embeddable representations within the optimization workflow. Through a dynamic mapping mechanism between geometric features and density fields, the filter evolves beyond a mere numerical tool for stabilizing optimization instabilities, becoming an active geometric feature modulation component in topology optimization. Since human design intent is directly embedded in the filter, the need for additional complex feature constraints is eliminated, significantly reducing optimization complexity. Numerical examples demonstrate the method's flexibility in generating structures with diverse geometric features, effectively facilitating human-machine interaction between design intent and structural mechanical performance.

Keywords: topology optimization, filtering techniques, geometry filter, geometric features, human-Informed, machine learning.

1 Introduction

As a powerful design tool, topology optimization has been widely adopted in engineering fields. Its core objective is to achieve optimal mechanical performance by seeking the most efficient material distribution under given load and boundary conditions. Due to its robust design capabilities, topology optimization has demonstrated significant application value in aerospace, automotive lightweighting, and biomedical implant design.

Among various topology optimization methods, the SIMP (Solid Isotropic Material with Penalization) method has become one of the most widely used approaches due to its clear physical interpretation and straightforward numerical implementation. However, in the SIMP method, the discrete distribution of material density may lead to alternating high and low-density arrangements in adjacent elements, forming a checkerboard pattern. This non-physical phenomenon can significantly reduce the engineering practicality of optimization results. Additionally, the optimization outcomes of the SIMP method may heavily depend on the finite element mesh discretization, where different mesh sizes or element types can yield entirely distinct topological structures. To mitigate these potential numerical instabilities, filtering techniques have been introduced into the topology optimization process. By smoothing the material density or sensitivity within the design domain, filtering effectively suppresses non-physical features caused by numerical instability, resulting in more coherent and practically viable optimization results. In early work, Sigmund et al. and Petersson et al. established sensitivity filtering schemes [1-3], where the actual sensitivities are filtered and the modified sensitivities are used to update the design. This approach has proven reliable for various problems. Bruns et al. and Bourdin et al. proposed density filters [4,5], which effectively suppress checkerboard patterns by performing weighted averaging of densities within the neighborhood of design variables. Bourdin et al. further developed convolution filtering methods [6], employing convolution operators to smooth material density, successfully addressing checkerboard effects and mesh dependency issues in topology optimization while enhancing the numerical stability and engineering applicability of design outcomes. These filtering schemes provide effective regularization tools for solving topology optimization problems.

However, the aforementioned studies merely proposed filters as "image processing" tools to suppress numerical instabilities in topology optimization, lacking more application-oriented geometric control capabilities. With deepening understanding, filtering techniques began to be employed for handling geometric constraints. For instance, Guest et al. achieved minimum feature size control in topology optimization by using nodal design variables and projection functions [7]. Their subsequent research further developed this concept [8], proposing a filtering-based maximum length scale constraint that provided maximum size control methods for topology optimization. Kawamoto et al. and Lazarov et al. introduced filters based on Helmholtz-type partial differential equations [9,10]. By implicitly defining filters through solving PDEs, they avoided the computational costs of neighborhood

searches in traditional filters. Later, Langelaar developed a filtering scheme tailored for additive manufacturing (AM) requirements from a filtering perspective [11], enabling overhang angle constraints in topology optimization. This filter ensured optimized designs met AM forming constraints through hierarchical geometric control. Vatanabe et al. further proposed a unified projection-based filtering technique [12] capable of handling multiple manufacturing constraints including minimum member size, minimum aperture, and symmetry.

However, as product design trends toward personalization and multifunctionality, the scope of topology optimization has long transcended pure mechanical metrics, gradually encompassing more subjective preferences such as geometric features, structural styles, and even design intuition. This not only demands topology optimization methods to achieve high standards in fundamental performance metrics but also expects the optimization process to facilitate human-computer interaction with designers while accommodating complex aesthetic styles and design features (e.g., biomimetic designs or aesthetic requirements) to match rapidly evolving multidimensional design needs. For example, texture replication techniques adopted texture-matching concepts [13-15], optimizing style as an appearance constraint function to guide stylistic emergence in topology optimization. Machine learning-assisted approaches proposed topology optimization methods for architecturally artistic designs [16-18], integrating structural performance with artistic elements by combining human design preferences. Furthermore, similar to introducing geometric patterns or styles into topology optimization, studies explored methods to incorporate more abstract human design intuition with automated topology optimization techniques [19-21], thereby enhancing design capabilities.

However, existing research primarily focuses on introducing desired structural features during the topology optimization process, without establishing intuitive quantification for design-challenging elements such as human intuition. This may lead to uncontrollable design styles in topology optimization and a lack of intuitive physical interpretation. It also exposes functional limitations in practical applications. The key issues are reflected in three aspects: 1) Limited functional positioning of Filters. Current Filters are still predominantly regarded as image post-processing techniques and have not been endowed with the capability to actively generate geometric structures. 2) Conflict between Filter introduction and design principles. Filters often emphasize ensuring local smoothness or meeting specific numerical constraints, which contradicts the need to generate structures with complex topologies and intricate geometric features. 3) Disconnection between Filters and design intent. In current topology optimization frameworks, high-level design elements such as stylistic features and human design intuition are typically introduced as explicit constraints into the topology model, remaining separate from Filters. This separation increases the complexity and solving difficulty of optimization problems. Therefore, exploring how to empower filters with the ability to directly generate geometric structures—embedding complex design requirements (e.g., geometric complexity, style preferences) implicitly into filter design or parametric representations (while maintaining controllability)—holds significant research value. This could expand the

design space of topology optimization and the functional scope of filters, elevating filters from traditional auxiliary image-processing tools to core components of topology optimization methods, achieving higher-level functional integration.

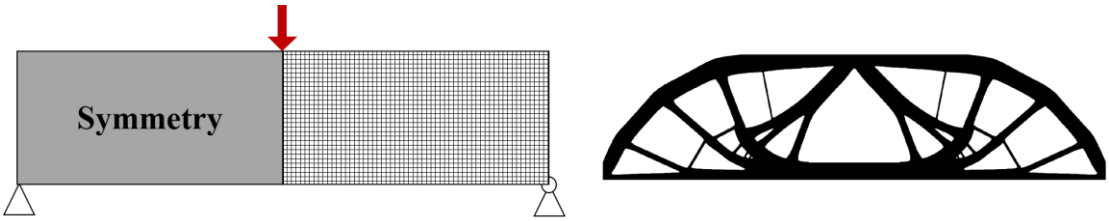
This study proposes a novel topology optimization method that incorporates diverse geometric information into filters through machine learning empowerment. The superposition and guidance of geometric elements are achieved via filters. In this method, "geometry" is regarded as a generalized concept aimed at capturing varied artificial design intents and reflecting different design considerations. Accordingly, its scope not only encompasses common stylistic geometries but also extends to biological structures, hand-drawn geometries, and other multidimensional geometric features that embody specific functions (e.g., mechanical biomimetics), design intuition (e.g., hand-drawn sketches), or more abstract, functionally oriented characteristics. This paper reshapes the filtering mechanism in topology optimization through machine learning, inherently embedding geometric shaping capabilities into the structural generation process. Unlike traditional post-processing or constraint superposition approaches, this method transforms design intent into an intrinsic property of a Geometry-Aware Filter. This filter no longer merely serves as a numerical stabilizer but evolves into a functional module that actively guides structural morphology evolution. By applying style-oriented guidance during iterations, the optimization process naturally converges to structural forms that combine preset geometric features with mechanical performance. This approach not only avoids the increased model complexity and computational costs associated with introducing additional constraints but also ensures high robustness and computational efficiency, ultimately achieving deep integration of generalized geometric features and structural performance.

The remainder of this paper is organized as follows: Section 2 will elaborate on the proposed topology optimization filtering method based on geometric information fusion, including the extraction and representation of geometric information, as well as specific strategies for integrating it into the filtering process. Section 3 will validate the method’s effectiveness through a series of numerical examples while exploring its applications with different types of geometric information, such as stylistic geometry, compliance geometry, and hand-drawn geometry, and analyzing how these diverse geometric influences affect optimization outcomes. Finally, Section 4 will summarize the main contributions of this work and outline future research directions.

2 Methods

2.1 Optimization Problem Modeling

The core idea of the SIMP method is to transform the discrete material distribution problem into a continuous density field optimization problem.



(a) A MBB example (b) Topology optimization result

Figure 1: Basic SIMP topology optimization framework

In SIMP, the design domain is discretized into a finite number of elements, with the material density of each element optimized as a continuous variable. The design domain is denoted as Ω , containing N elements. The material density of each element is represented by ρ_i , where $i = 1,2,\dots,N$. These density values are typically constrained between 0 and 1, $0 \le \rho_i \le 1$, where $\rho_i = 1$ indicates a fully solid element, while $\rho_i = 0$ denotes a void element.

Based on the above definitions, the topology optimization problem can be formulated as the following mathematical expression:

$$\begin{gathered} \min_{\rho} C(\rho) = U^T K U \\ V(\rho) = \sum_{i=1}^{N} \rho_i V_i \le V_{max} \\ 0 \le \rho_i \le 1, i = 1,2,\dots,N \\ KU = F \end{gathered} \tag{1}$$

The objective of the optimization problem is to minimize the structural compliance $C(\rho)$, where ρ represents the design variables, namely the material density of each element, to find the optimal material distribution. Here, U denotes the displacement vector of the structure, K is the stiffness matrix of the structure, which is a function of the material density ρ. The compliance function $C(\rho)$ measures the deformation energy of the structure under given loads; the smaller its value, the higher the structural stiffness. Minimizing compliance is a common optimization objective in topology optimization. The optimization process must satisfy the volume constraint $V(\rho) \le V_{max}$, where $V(\rho)$ represents the total volume of the structure, a function of the material density ρ . In the formula, V_i denotes the volume of the i -th element, and V_{max} is the preset maximum volume limit, ensuring the optimized structure does not exceed the predetermined material usage. The optimization problem satisfies the finite element equation $KU = F$, where F is the load vector.

The method proposed in this paper focuses on reshaping the filtering process in topology optimization by embedding geometric information to guide structural design styles. Since this innovation only affects the filtering stage, the core density-based topology optimization framework retains its original structure and solution mechanism, thereby ensuring the continuity and validity of subsequent analyses.

2.1.2 Geometry Feature-Based Filtering Method

To achieve geometry feature-guided topology optimization, the design domain is discretized into a rectangular grid composed of $n_i \times n_j$ elements. Each element is associated with a design variable density value $\rho_{(i,j)}$, where i and j represent the vertical and horizontal positions of the element, respectively. The optimization goal is to generate new density values $\tilde{\rho}_{(i,j)}$ from the design variables $\rho_{(i,j)}$ through the filtering process while incorporating geometric feature information.

In this method, the filtered density value $\tilde{\rho}_{(i,j)}$ of an element is influenced by the geometric feature density. The geometric feature density value $\phi_{(i,j)}$ of element (i,j) is defined as a combination of different geometric features:

$$\phi_{(i,j)} = g\left(\xi_{1(i,j)}, \xi_{2(i,j)}, \dots, \xi_{n(i,j)}\right) \tag{2}$$

In Equation (2), $\xi_{n(i,j)}$ represents the density value of the n -th geometric feature. As mentioned earlier, these "geometric features" are a broad concept that can encompass various forms such as stylistic geometry, compliance geometry (reflecting structural flexibility requirements), and even freehand geometry (capturing design intuition). Each feature characterizes a desired attribute of the structure from a specific dimension. The function g defines how these geometric features are integrated. In this study, for computational simplicity, the maximum function is adopted, namely:

$$\phi_{(i,j)} = max\left(\xi_{1(i,j)}, \xi_{2(i,j)}, \dots, \xi_{n(i,j)}\right) \tag{3}.$$

The maximum function simulates the "union" effect of multiple geometric features to some extent, effectively preserving the intensity of geometric features while remaining relatively simple and efficient in computation. Ultimately, the filtered density value $\tilde{\rho}_{(i,j)}$ of an element can be expressed as:

$$\tilde{\rho}_{(i,j)} = min\left(\rho_{(i,j)}, \phi_{(i,j)}\right) \tag{4}.$$

Equation (4) indicates that the filtered density value $\tilde{\rho}_{(i,j)}$ of element (i,j) depends on the minimum value between its design variable $\rho_{(i,j)}$ and the geometric feature density $\phi_{(i,j)}$. By adjusting different geometric features, fine control over the structural morphology can be achieved, allowing it to gradually exhibit predefined geometric characteristics during the topology optimization process.

To make the aforementioned filtering process differentiable, we need to apply smooth approximations to the non-smooth operators min and max in Equations (5) and (6). The smooth approximation of the min function can be expressed as:

$$\tilde{\rho} = smin(\rho, \phi) = \frac{1}{2}\left(\rho + \phi - \sqrt{(\rho - \phi)^2 + \epsilon} + \sqrt{\epsilon}\right) \tag{5}.$$

The smooth approximation of the max function can be expressed as:

$$\phi = smax\left(\xi_{1(i,j)}, \xi_{2(i,j)}, \dots, \xi_{n(i,j)}\right) = \left(\sum_{k=1}^{n} \xi_k^P\right)^{\frac{1}{P}} \tag{6}.$$

In Equations (5) and (6), ϵ and P are parameters controlling the accuracy and smoothness of the approximation. Adjusting these two parameters allows for a trade-off between computational efficiency and convergence during the optimization process.

Existing manufacturing-constraint-based filtering methods typically ensure structural manufacturability by considering the material density of underlying support units, with the core idea being to define support relationships through physical material distribution. The proposed method conceptually redefines the notion of "support" by introducing a new metric—geometric feature density $\phi_{(i,j)}$.

Specifically, while existing methods derive material support from the distribution of adjacent underlying units, this method anchors support relationships to whether a unit satisfies predefined geometric features. This shift from traditional physical material support to geometric feature support enables more sensitive and flexible responses to complex design intents.

2.2 Acquisition and Processing of Textured Geometric Information

2.2.1 Acquisition of Textured Geometric Information

The previous section introduced how to integrate geometric feature information into the topology optimization filtering process. This section will elaborate in detail on how to extract geometric feature information from input texture images and convert it into geometric feature density applicable to the filtering process.

To achieve this goal, the concept of texture synthesis is adopted. Texture synthesis aims to generate new textures with similar visual characteristics from a given example texture. This study utilizes the idea of texture synthesis to extract and generate geometric features that can guide topology optimization. Numerous existing works are dedicated to synthesizing textures from example inputs, and these methods can be broadly categorized into pixel-based and patch-based approaches. Local region-growing techniques synthesize textures pixel by pixel or patch by patch. Among them, patch-based methods typically succeed better in synthesizing high-quality textures, as they effectively preserve the global structure of the texture. On the other hand, pixel-based methods are more suitable for constrained synthesis, as they allow control over individual pixel values. This paper draws on patch-based texture synthesis methods, as they can generate high-quality texture information while maintaining the global structure of the texture.

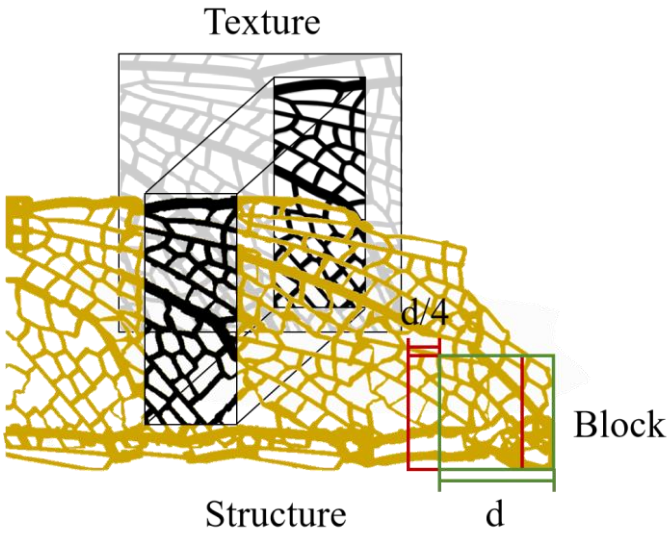


Figure 2: Method for extracting texture geometric features

The texture data is sourced from a library containing I geometric patterns, which consists of a series of PNG and OBJ files representing 2D and 3D geometric patterns, respectively. For the 2D case, the pixel values of the PNG files are directly interpreted as material density $\rho \in [0,1]$, expressed as $\rho(x,y) = \frac{I_{PNG}(x,y)}{255}$, where $\mathrm{I_{PNG}(x,y)}$ represents the pixel value of the PNG image at coordinates (x,y).

To extract geometric features, the entire design domain Ω is divided into several overlapping rectangular patches $B(\Omega_e)$. Each patch has dimensions of $d_x \times d_y$, where d_x and d_y are the horizontal and vertical sizes of the rectangular patch, respectively. The overlapping regions between patches are $d_x/4$ and $d_y/4$ in the horizontal and vertical directions, respectively. This overlap size setting follows the empirical practices of Hu and Kwatra et al. [22,23], who used similar strategies and achieved favorable results. For each patch $B(\Omega_e)$ in the design domain Ω , we attempt to find the most similar patch $\mathcal{I}_{\mathcal{B}}$ in the given input texture example image $\mathcal{J}$.

The squared norm is used to measure the distance between two patches. Specifically, for a patch B in the design domain and a patch $\mathcal{I}_{\mathcal{B}}$ in the example texture, their distance $D(B,\mathcal{I}_{\mathcal{B}})$ can be calculated by the following formula:

$$D(B,\mathcal{I}_{\mathcal{B}}) = \sum_{(i,j)\in B} w_{(i,j)} \left(B_{(i,j)} - h\left(\mathcal{I}_{(i,j)}\right)\right)^2 \quad (7).$$

Here, $B_{(i,j)}$ represents the density value of element i,j in patch B , and $J_{(i,j)}$ denotes the texture value of the corresponding element i,j in texture patch $J_{\mathcal{B}}$. By minimizing $D(B,\mathcal{I}_{\mathcal{B}})$, the most similar texture patch $\mathcal{I}_{\mathcal{B}}$ to the design domain patch B can be found in the input texture $\mathcal{I}$. Here, h is the Heaviside function, which maps the given texture to the interval $[0,1]$ and achieves a certain degree of binarization. Let $w_{(i,j)}$ be the weight, typically set to $w_{(i,j)} = B_{(i,j)}$. This weighting scheme focuses only on the texture similarity in material regions, i.e., areas with non-zero density

values. The texture patch with the smallest $D(B, \mathcal{J}_{\mathcal{B}})$ is identified as the optimal matching patch.

Compared to traditional appearance-based matching methods, performance-driven matching can more effectively enhance structural performance. However, directly evaluating the contribution of each texture patch to the overall structural performance involves high computational complexity. Conventional methods require placing each texture patch in the corresponding position within the structure and calculating compliance through finite element analysis to determine whether structural performance is improved. Although this approach offers high accuracy, it demands enormous computational effort and is inefficient. To reduce computational costs, this paper proposes a sensitivity-analysis-based method for evaluating texture contributions. Sensitivity reflects the impact of a slight change in element density on structural compliance variation, expressed mathematically as $\frac{dc}{d\rho}$, where dc denotes the change in compliance and dρ represents the infinitesimal change in density. By computing the density difference $\Delta\rho$ of a texture patch in the current structure, its influence on element density can be quantified. This, combined with sensitivity analysis, further approximates the patch's effect on structural compliance:

$$\Delta c = \frac{dc}{d\rho}\Delta\rho = \frac{dc}{d\rho}\left(B_{(i,j)} - h\left(J_{(i,j)}\right)\right) \tag{8}.$$

The texture block that minimizes the reduction in compliance is identified as the optimal matching block. While this method demonstrates high accuracy under small deformation conditions, it also provides effective reference under large deformation scenarios. Experimental results indicate that the sensitivity analysis-based approach can significantly enhance computational efficiency and yield structural design solutions with superior compliance. In practical applications, traditional appearance matching methods and performance-based matching methods each have their advantages, allowing flexible selection based on specific requirements. Appearance matching methods are suitable for scenarios with stringent structural appearance requirements, whereas performance-based matching methods are more appropriate for design tasks prioritizing structural performance optimization.

Through the aforementioned block matching process, each block B in the design domain finds its optimal matching block $\mathcal{J}_{\mathcal{B}}$ in the input texture $\mathcal{J}$. The density values of these matched texture blocks $\mathcal{J}_{\mathcal{B}}$ are assembled into one of the global texture geometric feature density fields, such as ξ_1 in the aforementioned Equation (7). This global texture geometric feature density field ξ_1 will serve as part of the geometric feature density field ϕ and be used in the filtering method described in the previous section to guide the topology optimization process. Overlapping regions between texture blocks are handled using a simplified direct coverage scheme. Subsequent numerical experiments demonstrate that this approach effectively preserves geometric feature representation while generating structures with excellent connectivity.

The above steps complete the conversion from an input texture image to a geometric feature density field, providing the necessary geometric information for subsequent topology optimization.

2.2.2 CNN-Based Rapid Texture Block Extraction

The previous section introduced a texture synthesis method based on block matching, which employs brute-force search to identify the most similar texture blocks in the exemplar texture for blocks in the design domain. However, this brute-force matching strategy incurs high computational costs, particularly when handling large-scale design domains and high-resolution textures, rendering it inefficient in practical applications. Although acceleration techniques such as fast matching algorithms, parallel processing, and random sampling can significantly improve matching speed in structural topology optimization and image editing, these methods still have limitations. On one hand, their matching speed remains insufficient for large-scale optimization problems; on the other hand, their approximate matching strategies may lead to loss of gradient information, which is detrimental to gradient-descent-based topology optimization algorithms.

In recent years, the rapid development of deep learning has provided a new paradigm for intelligently embedding and processing geometric information in topology optimization. Methods based on convolutional neural networks (CNNs) not only significantly accelerate related computations but, more crucially, can provide gradient information through backpropagation—this is essential for achieving geometry-guided topology optimization based on gradients. This study employs a CNN architecture whose core function is to learn and encode latent geometric features in local density fields. The network takes density blocks of the design domain as input and, through its learned ability to express geometric features, identifies and quantifies the most relevant geometric patterns from a predefined pattern library while outputting the gradients required for optimization.

To train the convolutional neural network (CNN), predefined geometric pattern libraries are first discretized into numerous texture blocks, each corresponding to a category in the pattern library. Subsequently, input data for network training are generated by applying random noise perturbations to these texture blocks, while the original, unperturbed texture blocks serve as the target output. For performance matching methods, random $\frac{dc}{d\rho}$ values can be generated as input, and the optimal texture blocks calculated using performance matching theory serve as output, thereby constructing the training database. The approach of randomly generating $\frac{dc}{d\rho}$ values not only enables rapid data generation, providing the network with diverse training samples and avoiding complex finite element analysis, but also exhibits greater robustness when handling complex geometric patterns.

The convolutional neural network (CNN) architecture adopted in this paper is relatively concise, primarily consisting of convolutional layers, batch normalization layers, and ReLU activation function layers stacked sequentially. The input layer of

the network receives image blocks of size $d_p \times d_p \times d_z$, where d_p represents the side length of the image block. In traditional 2D convolutional neural networks, d_z denotes the number of channels in the input image (e.g., RGB images have 3 channels), whereas in this study, d_z represents the thickness of the image along the z axis: for 2D problems, d_z is set to 1; for 3D data processing tasks, d_z indicates the number of sampling points along the depth direction.

Convolutional layers are used to extract local features from image blocks, batch normalization layers accelerate training and improve the network's generalization ability, and ReLU activation layers introduce nonlinearity to enhance the network's expressive power. Finally, the network outputs the density values of the texture blocks through a regression layer.

The training process employs the Adam optimization algorithm with hyperparameters set to a maximum of 100 epochs, an initial learning rate of 1×10^{-3} , and a mini-batch size of 32 to control the training. We also adopt a random shuffling strategy, shuffling the data before each epoch to further enhance training efficiency and stability. The entire training process is highly efficient, typically completing in under half a minute on a computer equipped with an RTX 4080 GPU. Once trained, the CNN network can be used for rapid texture patch extraction, providing essential information for subsequent geometric feature filtering. Notably, a corresponding pre-trained network can be generated for each geometric pattern, eliminating the need for retraining during each optimization, which further improves the computational efficiency of the method.

2.3 Numerical Techniques

2.3.1 Multi-Resolution Optimization

To reduce computational costs while ensuring texture clarity, this study employs a multi-resolution optimization strategy. The core idea is to start optimization at a low resolution and gradually increase it, thereby avoiding suboptimal local minima and accelerating the convergence of the optimization process.

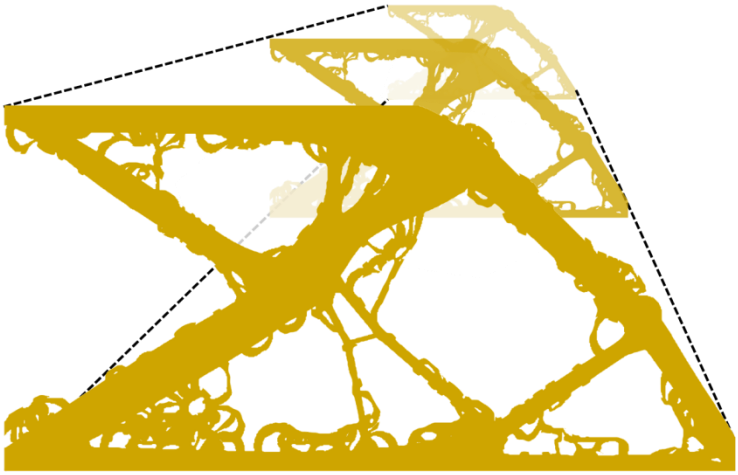


Figure 3: Multi-resolution optimization method

Specifically, this study adopts a three-resolution optimization strategy. The process begins with a low-resolution design domain Ω_1 and example texture I_1, where the size of Ω_1 is set to one-quarter of the target design domain. Note that under low-resolution optimization, the matched texture features are obtained by downsampling the high-resolution input texture example $\mathcal{J}$, rather than upsampling the low-resolution texture, which effectively preserves texture clarity. Simultaneously, the patch size associated with Ω_1 is also set to one-quarter of the high-resolution value. After completing optimization at low resolution, the results are upsampled using a bilinear interpolation scheme to serve as initial values for the next resolution level. This upsampling applies to both the design domain Ω and texture patches $\mathcal{I}_{\mathcal{B}}$. The process repeats, optimizing at progressively higher resolutions until the target resolution is reached, yielding the final optimized result.

2.3.2 Preprocessing and Progressive Introduction of Geometric Features

During the process of extracting geometric features, attention should be paid to potential gray areas in the input image. These gray areas correspond to intermediate density values in the density field, which may lead to gray-scale elements in the optimization results—a phenomenon generally undesirable in final binary structural designs. To prevent the generation of gray-scale elements, we employ the Heaviside projection function to preprocess geometric features, converting them into an approximately binary form. The Heaviside projection function not only achieves binarization but also preserves gradient information, facilitating the convergence of gradient-based optimization algorithms. The specific Heaviside projection function $h(\xi)$ is defined as follows:

$$h(\xi) = \frac{tanh(\beta_T T) + tanh\big(\beta_T(\xi - T)\big)}{tanh(\beta_T T) + tanh\big(\beta_T(1 - T)\big)} \tag{9}$$

Here, assuming ξ represents the geometric feature density value to be processed, T is the threshold (set to 0.5 in this study), and β_T is the parameter controlling the steepness of the Heaviside projection curve (empirically set to 25). This function maps the value of ξ to a range close to 0 or 1, thereby achieving an approximately binary effect. In addition to processing geometric features during optimization, binarization preprocessing can also be applied to the input texture image in the initial stage, both of which yield similar results.

However, introducing strong geometric feature constraints early in the optimization process may worsen problem behavior, making the optimization prone to falling into local optima. To avoid this issue, this study adopts a progressive introduction strategy for geometric features. Specifically, a control parameter is introduced to adjust the intensity of geometric features. The modified geometric feature density value $\tilde{\xi}$ is expressed as:

$$\tilde{\xi} = a + (b - a)h(\xi) \tag{10}$$

The geometric feature density $\phi_{(i,j)}$ in Equation (6) will also be calculated using $\tilde{\xi}$ processed by the Heaviside projection and progressive introduction strategy, i.e.:

$$\phi_{(i,j)} = smax\left(\tilde{\xi}_{1(i,j)}, \tilde{\xi}_{2(i,j)}, \dots, \tilde{\xi}_{n(i,j)}\right) \tag{11}$$

This formula remaps the geometrically filtered eigenvalues after Heaviside projection from the $[0,1]$ interval to the $[a, b]$ interval. During optimization, we fix $b = 1$ while gradually decreasing parameter a from 1 to 0. When $a = 1$, $\tilde{\xi}$ equals 1, meaning the geometric feature constraint is inactive, reducing the optimization problem to an unconstrained free optimization where all regions retain material. As a progressively decreases, the geometric constraint intensifies. When a finally reaches 0, $\tilde{\xi}$ reverts to the original Heaviside-projected value, allowing material retention only in regions with geometric feature support, achieving full geometric feature incorporation. This gradual introduction strategy enables the optimization algorithm to thoroughly explore the design space in early stages while progressively satisfying geometric constraints in later phases, effectively avoiding local optima and yielding high-quality designs.

Although the gradual introduction strategy is effective, a fixed a decrement rate may sometimes inadequately address dynamic complexities during optimization, potentially causing overly aggressive geometric feature incorporation that compromises stability. To address this, our study introduces an adaptive feedback control mechanism: it monitors real-time trends of key performance metrics (e.g., structural compliance) and dynamically adjusts the a reduction process. If objective function deterioration is detected, a reduction is paused to allow optimizer adaptation; conversely, if improvement occurs, a continues decreasing per schedule to steadily enhance geometric influence. This performance-based adaptive control not only refines geometric integration pacing but also improves optimization robustness and convergence quality, ultimately yielding structures with superior performance and morphology.

2.3.3 Sensitivity Analysis

To drive gradient-based optimization, the sensitivity of the objective function relative to design variables must be computed. This section details the derivation process of this sensitivity after implementing geometric feature filtering.

The structural compliance $C(\rho)$ can be expressed as:

$$C(\rho) = U^T K U \tag{12}$$

where U is the displacement vector, and K is the stiffness matrix−a function of material density ρ . The sensitivity of structural compliance to design variables is calculated via:

$$\frac{dC}{d\rho_{(i,j)}} = -U^T \frac{dK}{d\rho_{(i,j)}} U \tag{13}$$

.

Here, $\frac{dK}{d\rho_{(i,j)}}$ represents the derivative of the stiffness matrix with respect to the design variable $\rho_{(i,j)}$. Since geometric feature information is introduced during the filtering process, the influence of filtering on sensitivity must be considered. According to the chain rule, the derivative of stiffness matrix K with respect to design variable $\rho_{(i,j)}$ can be expanded as:

$$\frac{dK}{d\rho_{(i,j)}} = \frac{dK}{d\tilde{\rho}_{(i,j)}} \frac{d\tilde{\rho}_{(i,j)}}{d\rho_{(i,j)}} \tag{14}$$

$\frac{dK}{d\tilde{\rho}_{(i,j)}}$ denotes the derivative of the stiffness matrix with respect to the filtered density $\tilde{\rho}_{(i,j)}$, which can typically be obtained through finite element analysis; furthermore, $\frac{d\tilde{\rho}_{(i,j)}}{d\rho_{(i,j)}}$ can be expressed as:

$$\frac{d\tilde{\rho}_{(i,j)}}{d\rho_{(i,j)}} = \frac{\partial\tilde{\rho}_{(i,j)}}{\partial\rho_{(i,j)}} + \frac{\partial\tilde{\rho}_{(i,j)}}{\partial\phi_{(i,j)}} \frac{d\phi_{(i,j)}}{d\rho_{(i,j)}} \tag{15}$$

Although the computation of $\phi_{(i,j)}$ involves all densities $\rho_{(k,l)}$ within the rectangular block B where element i,j is located (where $(i,j) \in B$), $\tilde{\rho}_{(i,j)}$ is solely determined by $\rho_{(i,j)}$ and $\phi_{(i,j)}$. Therefore, when calculating $\frac{\partial\Phi_{(i,j)}}{\partial\rho_{(i,j)}}$, only the influence of density value $\rho_{(i,j)}$ on $\phi_{(i,j)}$ needs to be considered, without accounting for other element densities. Further expanding, the computation of $\frac{\partial\tilde{\rho}_{(i,j)}}{\partial\rho_{(i,j)}}$ involves the differentiation of the min function. We employ the smooth function smin to replace the min operator. Thus, the partial derivatives of $\tilde{\rho}_{(i,j)}$ with respect to $\rho_{(i,j)}$ and $\phi_{(i,j)}$ are respectively:

$$\frac{\partial \tilde{\rho}_{(i,j)}}{\partial \rho_{(i,j)}} = \frac{1}{2}\left(1 - \frac{\rho_{(i,j)} - \phi_{(i,j)}}{\sqrt{\left(\rho_{(i,j)} - \phi_{(i,j)}\right)^2 + \epsilon}}\right) \tag{16}$$

$$\frac{\partial \tilde{\rho}_{(i,j)}}{\partial \phi_{(i,j)}} = \frac{1}{2}\left(1 + \frac{\rho_{(i,j)} - \phi_{(i,j)}}{\sqrt{\left(\rho_{(i,j)} - \phi_{(i,j)}\right)^2 + \epsilon}}\right) \tag{17}$$

Whereas $\phi_{(i,j)}$ is a function of $\xi_{k(i,j)}$. When geometric features include texture characteristics ξ_1 obtained through appearance matching, the influence of $\tilde{\xi}_1$ must be considered. By the chain rule, the derivative of $\phi_{(i,j)}$ with respect to design variable $\rho_{(i,j)}$ can be expressed as:

$$\frac{d\Phi_{(i,j)}}{d\rho_{(i,j)}} = \frac{\partial \Phi_{(i,j)}}{\partial \tilde{\xi}_{1(i,j)}} \frac{\partial \tilde{\xi}_{1(i,j)}}{\partial \xi_{1(i,j)}} \frac{\partial \xi_{1(i,j)}}{\partial \rho_{(i,j)}} + \sum_{k=2}^{n} \frac{\partial \Phi_{(i,j)}}{\partial \tilde{\xi}_{k(i,j)}} \frac{\partial \tilde{\xi}_{k(i,j)}}{\partial \xi_{k(i,j)}} \frac{\partial \xi_{k(i,j)}}{\partial \rho_{(i,j)}} \tag{18}$$

Sketches, flexibility solutions, and other geometric features do not change with variations in density; therefore, $\frac{\partial \xi_{k(i,j)}}{\partial \rho_{(i,j)}}$ is 0. Furthermore, the second term in the equation is 0. Considering the first term in the above equation, we use the smax function for smooth approximation, and the partial derivative of $\phi_{(i,j)}$ with respect to $\xi_{1(i,j)}$ is:

$$\frac{\partial \phi_{(i,j)}}{\partial \xi_{1(i,j)}} = \xi_{1(i,j)}^{P-1} \left(\sum_{l=1}^{n_S} \xi_{l(i,j)}^{P}\right)^{\frac{P}{P-1}} \tag{19}$$

Based on the definition of $\tilde{\xi}_1$, we have:

$$\frac{\partial \tilde{\xi}_{1(i,j)}}{\partial \xi_{1(i,j)}} = a + (b-a)\frac{dh\left(\xi_{1(i,j)}\right)}{d\xi_{1(i,j)}} \tag{20}$$

$$= a + (b-a)\frac{\beta_T \left(sech\left(\beta_T\left(\xi_{1(i,j)} - T\right)\right)\right)^2}{tanh(\beta_T T) + tanh\left(\beta_T(1-T)\right)} \tag{21}$$

where sech denotes the hyperbolic secant function.

This study primarily focuses on obtaining texture features through CNN; thus, $\frac{\partial \xi_{1(i,j)}}{\partial \rho_{(i,j)}}$ can be derived via backpropagation in CNN. This gradient information is crucial for calculating the sensitivity of geometric feature density with respect to design variables. This term is often ignored and set to 0 in traditional methods[23]. In the proposed method, CNN enables precise computation of this gradient, thereby improving optimization accuracy. However, it is worth noting that for performance-

matching methods, this term must still be neglected as 0 due to the inability to establish the relationship between $\xi_{1(i,j)}$ and $\rho_{(i,j)}$, following conventional practice.

3 Numerical Examples

In the numerical examples of this study, the MATLAB programming environment is adopted, implemented based on the classic 99-line topology optimization code [24], and solved using the Optimality Criteria (OC) method. All design domains are assumed to consist of Poisson's ratio $v = 0.3$, Young's modulus $E_0 = 1$, and minimum Young's modulus $E_{min} = 1 \times 10^{-4}$. To enhance the precision and efficiency of optimization results, a three-resolution optimization strategy is employed, progressively optimizing at low, medium, and high resolutions. The convergence criterion is defined as a design variable change of less than 1×10^{-3} or a relative change in the objective function of less than 1×10^{-4} . The maximum iteration steps for each resolution are set to 100, which experiments confirm is sufficient to ensure convergence and stable topology optimization results. Initial design variables are uniformly distributed with a density value of $\rho_0 = 0.5$. A density filter is applied with a radius of $r_{min} = 2.0$ to avoid checkerboard patterns and ensure smooth optimization results.

3.1 Benchmark Examples

This section demonstrates the effectiveness of the proposed method through three comparative examples, including classical simply-supported beam optimization, texture geometry-driven optimization, and sketch-based mechanics intuition-driven optimization. Figure 4a illustrates the basic setup of the examples: the left end of the classical simply-supported beam is fixed, while the right end is subjected to a vertical displacement constraint, with a load magnitude of $F = 1$ applied at the center of the upper structure. For simplified computation, half of the structure is analyzed. The design domain is discretized into meshes of 40×80、80×160 and 160×320 under low, medium, and high resolutions, respectively, with a volume fraction set to $V_{max} = 0.4$.

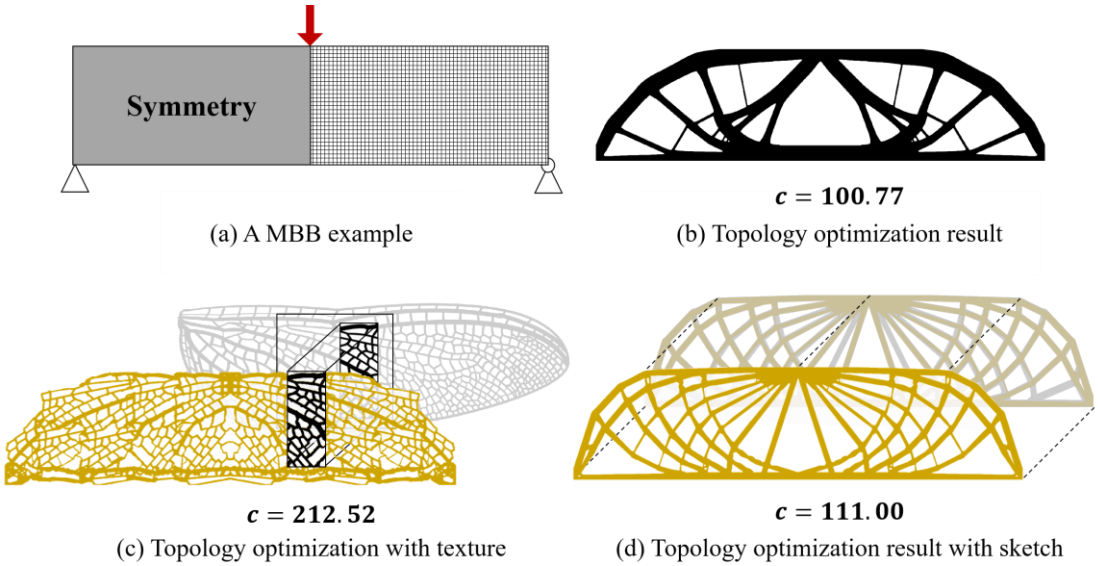


Figure 4: Comparison of optimization results for classical simply-supported beam, texture geometry-driven optimization, and mechanics intuition-driven optimization

Figure 4b presents the results of pure compliance optimization, which incorporates no geometric information, with the sole objective of minimizing compliance. The final compliance value is $c = 100.77$. This result serves as a benchmark for comparison with other examples.

Figure 4c displays the topology optimization results driven by texture geometry. The outcome shows that the global structure is generated under texture geometry guidance, but due to significant design space constraints, the compliance value increases markedly to $c = 212.52$. This example verifies the strong constraining effect of texture geometric features on optimization results.

Figure 4d shows the optimization results driven by a sketch, which was drawn by the designer based on mechanics intuition following a Michell truss layout. By incorporating mechanics intuition, the compliance value of the optimized result improves compared to pure texture-driven optimization, reaching $c = 111.00$. The use of filtering technology ensures the optimized result strictly adheres to the predefined geometric features, demonstrating the method's effective control over geometric characteristics.

Notably, some local protrusions appear on the structural boundary in Figure 4c. This occurs because the optimization problem can sometimes unavoidably converge to local optima, preventing further exploration of better material distributions. Additionally, the optimized results exhibit almost no fractured structures, as the sole objective is minimizing compliance, eliminating the need to sacrifice continuity for other constraints.

The comparison of these three examples fully validates the flexibility of the proposed method in integrating texture geometry, mechanics intuition, and topology

optimization, enabling the generation of structurally feasible designs while satisfying geometric feature constraints.

3.2 Compliance and Texture Combined Optimization

To validate the effectiveness of the proposed method in fusing multiple geometric features, the classical short beam problem illustrated in Figure 5a was selected as a test case. The left boundary of the beam was fully fixed, with a vertically downward concentrated load of magnitude 1 applied at the lower right corner. Compliance was incorporated as one of the geometric features, combined with texture features. Specifically, optimized compliance solutions under different volume fractions (0.3, 0.25, 0.2) were used as compliance-based geometric features, onto which the floral texture geometric feature shown in Figure 2b was superimposed. During optimization, the upper limit of the total volume fraction was set to 0.4.

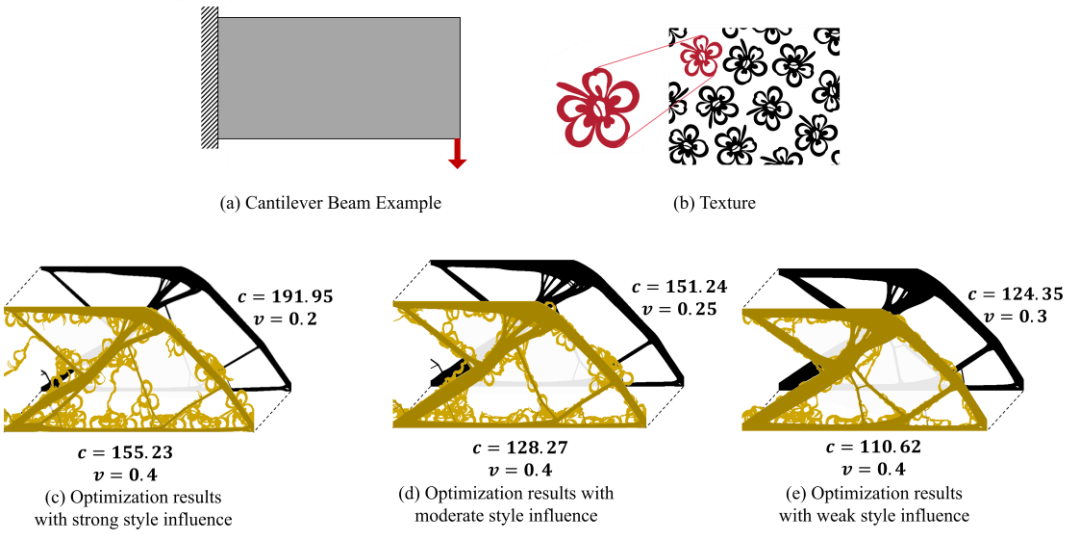


Figure 5: Optimization results combining compliance and texture

The optimization results demonstrate that as the volume fraction of compliance solutions increases, texture features gradually emerge and occupy more areas within the structure. This indicates that the proposed method effectively integrates texture features into the optimization process, achieving varying degrees of stylized expression under different volume fractions. Since compliance-optimized solutions are incorporated as geometric features, the mechanical performance of the optimized results surpasses that of single-texture or single-sketch-guided outcomes, though remains inferior to pure compliance-optimized solutions. This aligns with expectations, as introducing texture features inevitably compromises structural mechanical performance to some extent. Moreover, structural compliance progressively increases with the proliferation of texture features—an intuitive outcome given that texture features typically exhibit weaker mechanical properties, and their increased presence necessarily elevates overall compliance.

Further observation reveals that texture features predominantly appear near structural support regions. This occurs because the optimization objective is to

minimize compliance, prompting the structure to seek mechanically optimal solutions near supports under geometric constraints. The introduction of texture features alters the search path of the objective function, guiding the optimization algorithm to preserve texture geometry while satisfying mechanical performance requirements.

Compared to existing constraint-based methods, the proposed approach integrates texture features into the optimization process through filtering techniques while optimizing under conditions of minimal compliance and volume constraints. Conventional methods typically treat texture features as additional constraints that must be strictly satisfied during optimization, which may lead to potential conflicts between texture features and compliance minimization objectives. For instance, relevant studies indicate that after introducing texture constraints, the fulfillment of volume constraints may be compromised, and optimization efficiency can significantly decrease [15]. To satisfy texture constraints, structures may require additional material or morphological changes in certain areas, thereby affecting mechanical performance and potentially generating structural forms detrimental to stiffness. In contrast, our method naturally incorporates texture features into the optimization process via filtering techniques without introducing additional constraint functions. Under the premise of meeting volume constraints, the sole optimization objective is minimizing compliance, allowing texture features to be fully utilized as structural components that bear partial mechanical loads while maintaining good structural connectivity. This approach not only avoids material waste caused by forcibly satisfying constraints but also achieves better balance between mechanical performance and aesthetic characteristics. Furthermore, by eliminating the need to handle complex constraints, our method significantly improves computational efficiency and the robustness of the optimization process.

4 Conclusions and Contributions

This study proposes a novel topology optimization method based on filtering techniques, aiming to deeply integrate geometric features into the structural design process. Departing from conventional topology optimization strategies that treat aesthetic or geometric constraints as add-ons, this method embeds predefined geometric features into the filtering process from the initial optimization stage, transforming them into intrinsic drivers of structural generation. These geometric features can either originate from manually designed key geometric elements or be extracted as stylized patterns from textures. Through innovation in traditional filtering techniques, this research organically combines geometric patterns with the density field filtering process, guiding the optimization algorithm to follow predefined geometric trajectories while searching for optimal material distributions, thereby constructing structural forms that possess both excellent mechanical performance and distinctive aesthetic characteristics. The method not only enables fully texture-driven structural generation but also flexibly integrates textures with stiffness-optimized results or other conceptual geometries based on designer intuition, creating diverse hybrid structural forms and achieving precise control over geometric features. Since aesthetic elements are embedded early in the optimization, the specified geometric features naturally blend into the structure itself, resulting in fluid and organic aesthetic

qualities that avoid the disharmony introduced by post-hoc manual adjustments or additional constraints. Compared to traditional methods, this approach eliminates the need for complex external constraint functions, significantly improves computational efficiency and robustness, effectively mitigates the risk of converging to local optima, and demonstrates the capability to flexibly generate structures with diverse geometric features. While meeting mechanical performance requirements, it naturally endows structures with unique aesthetic styles, achieving a high degree of unification between design intent and engineering feasibility.

However, this study still has certain limitations that need further improvement. Firstly, the current deep learning technology employed is primarily based on simple matching principles and has not yet fully explored the deep semantic information of textures. Future research could consider introducing more advanced texture fusion extraction techniques, such as VGG19-based deep learning models, to capture richer texture features. Secondly, compared to methods that treat texture style as an explicit constraint, our approach, while more intuitive in style control, still has room for improvement in control precision. Currently, the balance between style and flexibility is qualitatively adjusted by optimizing the volume fraction through flexibility regulation, which cannot achieve fine-grained style control. Additionally, the current geometric feature extraction method is mainly trained on a limited set of texture samples. Future work could further expand the corresponding neural network models to achieve adaptive adjustment of texture pattern size and orientation by learning from massive samples, which may require constructing deeper and more complex network architectures. Finally, in the current version, sketches are provided by designers in a single pass, whereas in actual design workflows, iterative refinement and real-time adjustments are often necessary. Therefore, enhancing user interaction capabilities to allow designers to adjust geometric features in real-time during the optimization process will be an important direction for future research.

Acknowledgements

The financial support from the Foundation for Innovative Research Groups of the National Natural Science Foundation (11821202), and the National Natural Science Foundation (12272075) are gratefully acknowledged.

Conflict of interest

The authors declared no potential conflicts of interest with respect to the research, author-ship, and/or publication of this article.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] Sigmund, O., 1994. Design of material structures using topology optimization (Doctoral dissertation, Technical University of Denmark).

[2] Sigmund, O., 1997. On the design of compliant mechanisms using topology optimization. Journal of Structural Mechanics, 25(4), pp.493-524. https://doi.org/10.1080/07365909708908587

[3] Sigmund, O. and Petersson, J., 1998. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. Structural optimization, 16, pp.68-75. https://doi.org/10.1007/BF01197754

[4] Bruns, T.E. and Tortorelli, D.A., 2001. Topology optimization of non-linear elastic structures and compliant mechanisms. Computer methods in applied mechanics and engineering, 190(26-27), pp.3443-3459. https://doi.org/10.1016/S0045-7825(00)00278-4

[5] Bourdin, B., 2001. Filters in topology optimization. International journal for numerical methods in engineering, 50(9), pp.2143-2158. https://doi.org/10.1002/nme.116

[6] Bourdin, Blaise. "Filters in topology optimization." International journal for numerical methods in engineering 50.9 (2001): 2143-2158. https://doi.org/10.1002/nme.116

[7] Guest, J.K., Prévost, J.H. and Belytschko, T., 2004. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. International journal for numerical methods in engineering, 61(2), pp.238-254. https://doi.org/10.1002/nme.1064

[8] Guest, J.K., 2009. Imposing maximum length scale in topology optimization. Structural and Multidisciplinary Optimization, 37, pp.463-473. https://doi.org/10.1007/s00158-008-0250-7

[9] Kawamoto, A., Matsumori, T., Yamasaki, S., Nomura, T., Kondoh, T., & Nishiwaki, S. (2010). Heaviside projection based topology optimization by a PDE-filtered scalar function. Structural and Multidisciplinary Optimization. doi:10.1007/s00158-010-0562-2

[10] Lazarov B, Sigmund, O. (2010) Filters in topology optimization based on Helmholtz-type differential equations. Int J Numer Methods Eng, 81(5), 607-627.

[11] Langelaar, Matthijs. "An additive manufacturing filter for topology optimization of print-ready designs." Structural and multidisciplinary optimization 55 (2017): 871-883.

[12] Vatanabe, Sandro L., et al. "Topology optimization with manufacturing constraints: A unified projection-based approach." Advances in Engineering Software 100 (2016): 97-112.

[13] Martínez, J., Dumas, J., Lefebvre, S., & Wei L.-Y. (2015). Structure and appearance optimization for controllable shape design. ACM Transactions on Graphics, 34(6), 12. https://doi.org/10.1145/2816795.2818101

[14] Hu J., Li M., & Gao, S. (2019). Texture-guided generative structural designs under local control. Computer-Aided Design, 108, 1 - 11. https://doi.org/10.1016/j.cad.2018.10.002
[15] Navez, T., Schmidt, M.-P., Sigmund, O., & Pedersen C. B. W. (2022). Topology optimization guided by a geometrical pattern library. Structural and Multidisciplinary Optimization, 65(108), 1 - 15. https://doi.org/10.1007/s00158-022-03197-x
[16] Zhang, Weisheng, et al. 2023. " Machine-learning Assisted Topology Optimization for Architectural Design with Artistic Flavor." Comput. Methods Appl. Mech. Engrg. 413: 116041. https://doi.org/10.1016/j.cma.2023.116041.
[17] Gatys, Leon A., Alexander, S. Ecker, and Matthias Bethge. 2016. "Image Style Transfer Using Convolutional Neural Networks." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2414-2423. https://doi.org/10.1109/CVPR.2016.84
[18] Vulimiri, P.S., Deng, H., Dugast, F., Zhang, X. and To, A.C., 2021. Integrating geometric data into topology optimization via neural style transfer. Materials, 14(16), p.4551. https://doi.org/10.3390/ma1416455
[19] Wang, Y., et al. (2023). From Computer-Aided Design (CAD) toward Human-Aided Design (HAD): An Isogeometric Topology Optimization Approach. Engineering. https://doi.org/10.1016/j.eng.2022.07.013
[20] Zhang, W., et al. (2024). Machine learning powered sketch aided design via topology optimization. Computer Methods in Applied Mechanics and Engineering, 419, 116651. https://doi.org/10.1016/j.cma.2023.116651
[21] Ha, D. Q., & Carstensen, J. V. (2023). Human-Informed Topology Optimization: Interactive application of feature size controls. Structural and Multidisciplinary Optimization, 66, 59. https://doi.org/10.1007/s00158-023-03512-0
[22] Kwatra, V., Essa, I., Bobick, A. and Kwatra, N., 2005. Texture optimization for example-based synthesis. In ACM Siggraph 2005 Papers (pp. 795-802).
[23] Hu, J., Li, M. and Gao, S., 2019. Texture-guided generative structural designs under local control. Computer-Aided Design, 108, pp.1-11.
[24] Sigmund, O., 2001. A 99 line topology optimization code written in Matlab. Structural and multidisciplinary optimization, 21(2), pp.120-127. https://doi.org/10.1007/s001580050176