Topology optimization guided by a geometrical pattern library

Navez, Tanguy; Schmidt, Martin-Pierre; Sigmund, Ole; Pedersen, Claus B. W.

Published in:
Structural and Multidisciplinary Optimization

Link to article, DOI:
10.1007/s00158-022-03197-x

Publication date:
2022

Document Version
Peer reviewed version

Topology Optimization Guided by a Geometrical Pattern Library

Tanguy Navez[1, 3, x], Martin-Pierre Schmidt[2, x], Ole Sigmund[3], and Claus B.W. Pedersen[4]

[1]Dept. of App. Math. and Comp. Sci., Technical University of Denmark, Lyngby, Denmark
[2]Lab. of Mathematics, National Institute of Applied Sciences, Rouen, France
[3]Dept. of Mech. Eng., Technical University of Denmark, Lyngby, Denmark
[4]Willhöden 30, 22587 Hamburg, Germany
[x]Equal contribution

March 14, 2022

Abstract

This work proposes an approach for structural Topology Optimization enforcing geometrical features on optimized designs using a predefined library of geometrical patterns. The approach applies a density-based Topology Optimization subject to a geometrical constraint guiding the design toward shapes matching the geometrical features found in the predefined pattern library. Multiple distance measures and suitable matching algorithms are studied to calculate local mappings between the design in each optimization iteration and the pattern library. An aggregated appearance constraint is evaluating the pattern matching. The gradient of the appearance constraint, objective function and other constraints are applied in a gradient-based optimization scheme. A parameter for the appearance constraint dictates how closely the design should match the patterns defined in the library. The convergence behavior is studied on a variety of 2D and 3D optimization scenarios. The formulation is also applied to design variables controlling the material orientations alongside the material density as well as other optimization objectives such as stress minimization.

Keywords: Topology Optimization, Geometrical Pattern Library, Appearance Constraint, Fast Matching Algorithm

1 Introduction

Additive Manufacturing has seen tremendous popularity in recent years and calls upon new 3D modeling and design approaches. The increased manufacturing freedom motivates generative design methods such as structural Topology Optimization (Bendsøe and Sigmund, 2003). Traditionally, structural topology optimization targets physical and mechanical performance such as stiffness, strength, buckling, heat conduction, *etc.* (Deng et al., 2021), (Gao and Ma, 2015). In addition, geometrical constraints can improve the regularity or manufacturability of the optimized designs (Wu et al., 2016), (Langelaar, 2017), (Langelaar, 2016), (Hoffarth et al., 2017).

In certain fields such as architecture and consumer product design, aesthetic considerations accompany the mechanical performance and manufacturing requirements. Today, such aesthetic geometrical features are rarely considered in the topology optimization process. Visual geometrical feature enhancement operations are often handled manually by trial and error or as a post-processing step after the topology optimization. These manual operations can greatly deteriorate the obtained optimized mechanical performance. Therefore, it is preferable to simultaneously optimize the mechanical performance and aesthetics of the design. Additionally, control over geometrical patterns appearing in the design can also serve to enhance its overall mechanical performance, for example by producing oriented microstructures (Wu et al., 2021). Likewise, it may allow to encourage certain desirable geometrical features for manufacturing or avoid pathological defects in the design.

Overall, the challenge lies in guiding the design using a geometrical library mathematically and algorithmically to describe concepts like *style*, *appearance* or *aesthetics* of a given design. Most methods attempting to tackle these concepts are in the field of Computer Graphics using a *guide by example* paradigm (Wei et al., 2009). Similarly, this work describes a formulation where the optimized design is forced to exhibit geometrical features present in a user-defined library of patterns. This formulation relies on a matching algorithm and the computation of a distance function between the design and the pattern library, as well as its first order derivatives required for efficiently applying gradient-based optimization schemes. Note that the kind of geometrical pattern matching targeted in this paper is both size and orientation sensitive. Fig. 1 illustrates the optimization approach for a simple problem where the design is guided by a library of regular grid-like patterns.

This article is organized as follows. Section 2 describes related work upon which some components of the present work are based. Section 3 describes the mathematical algorithms used to define and solve the topology optimization subject to an appearance constraint for a predefined geometrical pattern library. Section 4 shows the approach on a variety of 2D and

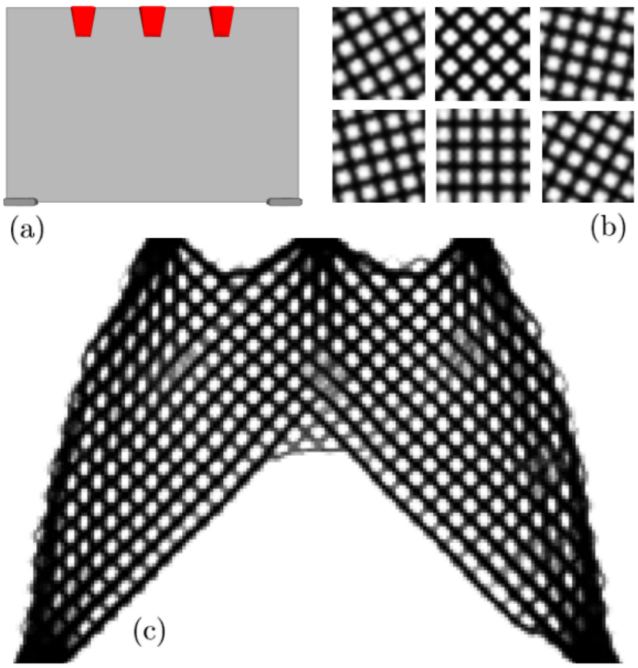

Figure 1: (a) 2D bridge optimization having three loaded regions at the top and two clamped regions at the bottom corners. (b) The library of six different geometrical patterns. (c) The optimized design exhibiting geometrical features of the pattern library.

3D scenarios and discusses the results including the optimization iteration convergence. Section 5 extends the approach by applying it to optimization for orthotropic materials also having the angles as additional design variables. Finally, Section 6 concludes on the present approach and discusses potential future work.

2 Related work

Synthesizing images originates from the field of *texture synthesis* creating or extending rectangular textures from small examples. Readers may refer to (Wei et al., 2009) for a review of such methods. The problem of synthesizing a new image resembling a so-called *exemplar* is addressed using minimization of an energy function. Pixel values of the output image are the variables being modified through the optimization of an *Appearance* energy (see (Kwatra et al., 2005)). This approach is adopted in our work to quantify the appearance of the generated design in relation to provided predefined geometrical patterns.

Patch-based Synthesis is an active field of research in Computer Graphics where images are synthesized by copying and assembling small regions or patches from within an exemplar. A comprehensive review of patch-based texture synthesis methods up to 2017 is available in (Barnes and Zhang, 2016). This type of local approach operates on compact regions, and is therefore easily coupled with density based Topology Optimization. The main stage in most patch-based synthesis methods is the matching phase finding suitable patches at a given region from a geometrical pattern library. Although search in the design domain can be done by exhaustively sampling the search space, this strategy is computationally too expensive for any non-trivial applications. More efficient approximation algorithms are preferred, such as PatchMatch (Barnes et al., 2009) which is adapted in our study and discussed later in the paper. These techniques used in Texture Synthesis are easily translated to density-based Topology Optimization by describing the pixel values as analogous to the design variables being defined by the finite element discretizing of the design domain.

We propose to guide the appearance optimization from a collection of several patterns instead of only one pattern. This idea is derived from the WaveFunctionCollapse Github project (Gumin, 2016) generating procedural coherent bitmaps from tilemaps of several patterns by constraining their positions on a fixed finite element grid. The tilemaps are constrained by edge conditions. For example, a tile whose right side is green can only be positioned to the left of a tile whose left side is green. The idea is to allow the algorithm to choose between several inputs, which are more or less locally adapted to the generated structure. However, it is important to note that the hard constraint of conformity at the interface of each patch was quickly abandoned in this work as a smooth transition through overlapping patches is better suited for gradient-based optimization.

A few publications have considered both texture synthesis and 3D manufacturing. The strength of a design is improved as a post-processing step after its modelling by locally modifying the thinnest parts of the geometry in (Stava et al., 2012). Geometric patterns can be synthesized along the surface of a 3D object while ensuring it stays printable as shown in (Dumas et al., 2015). However, this last approach does not take into account the amount of material and tends to fill the entire design domain. Fine filigree is created along the surface of any 3D design to resemble a 2D exemplar in (Chen et al., 2016).

Finally, (Martínez et al., 2015) proposed a formulation for optimizing the similarity of a 2D design or an assembly consisting of 2D designs with a single pattern while applying constraints on mass and minimizing the objective being the sum of compliance and pattern similarity. This approach seems to offer the strongest coupling between aesthetics and mechanical performance as both objectives are continually reevaluated during the design optimization process.

2.1 Contribution

The present article is based on the previous work (Martínez et al., 2015) considering a single geometrical pattern. Our approach is generalized to operate in 3D having a library of multiple geometrical patterns that the optimization can pick from and with a modified distance metric for improved numerical handling of low-density regions. Furthermore, we demonstrate the numerical robustness of this formulation by applying it to more challenging design optimization such as stress minimization and angle design variables. We also present an al-

ternative formulation using a soft-min aggregation letting the optimization choose to reproduce a pattern from the library only in the location having the smallest impact on the mechanical performance. Finally, we demonstrate the formulation on an optimization problem having an anisotropic constitutive material model and where the design variables are not only the material density but also its local orientation.

3 Mathematical Model

This section presents the mathematical modeling for topology optimization subject to an additional appearance constraint. We describe how we measure the local similarity of a generated structure to a geometrical pattern library and then we formulate a differentiable measure of appearance. Additionally, we describe a matching assignment algorithm determining which patterns from the library should be applied in a given design.

3.1 Topology Optimization Formulation

The aim of topology optimization is to find an optimal distribution of material within a given design domain according to a set of objective and constraints. The topology optimization formulation used in our study is based on the density-based SIMP approach (Bendsøe and Sigmund, 1999). In most of the following examples, we are minimizing the compliance as objective function subject to a mass constraint and the proposed Appearance constraint. In the later sections, we also show examples where we apply an objective function to minimize the maximum peak stress or optimize the local orientations of anisotropic material alongside density design variables.

The design domain Ω is subdivided into a regular grid of quadrilateral or hexahedral finite elements i in 2D and 3D, respectively. This results in the following optimization problem:

$$\begin{aligned} arg_{\rho} min \quad & J(\rho) = \mathbf{f}^T \mathbf{u} \\ s.t. \quad & \mathbf{K}(\rho)\mathbf{u} = \mathbf{f} \\ & 0 \leq \rho_i \leq 1 \quad \forall i \in \Omega \\ & G(\rho) = \frac{1}{|\Omega|} \frac{\sum_{i \in \Omega} \rho_i}{G^*} \leq 1 \\ & A(\rho) \leq 1 \end{aligned} \tag{1}$$

The effective Young modulus Y of each element i for the assigned isotropic material Y_0 is determined by the power law approach $Y(\rho_i) = \rho_i^p Y_0$ where the penalisation exponent $p = 3$. The structural equilibrium associates the global stiffness matrix $\boldsymbol{K}$, the nodal force vector $\boldsymbol{f}$ and the nodal displacement vector $\boldsymbol{u}$. Then we can solve for $\boldsymbol{u}$ and calculate the objective function $J(\rho)$. The densities ρ are bounded between 0 and 1 by box constraints. The volume fraction $G(\rho)$ is constrained by G^*. As one of our contributions, we add the appearance constraint $A(\rho)$ which is defined in section 3.2.

A filter is applied to the sensitivities (Sigmund, 1997) to improve numerical stability, reduce erroneous checkerboard patterns (Díaz and Sigmund, 1995) and introduce a minimum length scale in the optimization. We use the Method of Moving Asymptotes (MMA) (Svanberg, 1987) gradient-based mathematical programming approach to update the design variables. Once the design variables have been updated then the algorithm loops back to the evaluation of the objective and constraint functions if the stopping criteria is not fulfilled.

3.2 Appearance Constraint

In this section, we define the Appearance constraint $A(\rho)$. The appearance value is modeled as a function of the density field ρ and a predefined set of geometrical patterns I. A lower appearance value indicates small difference between ρ and the patterns I. In other words, it means that every small patch of ρ is *visually* very similar to at least one patch of a geometrical pattern in I. In the following, we describe our choices of representation and the patch discretization. Afterwards, we describe the distance metric used to compare two patches. Then, the matching algorithm for computing a mapping between patches of Ω and I is described. Finally, the aggregated appearance function and its partial derivatives are defined with respect to the design variables.

3.2.1 Geometrical Pattern Library

We define a library consisting of $|I|$ geometrical patterns. In the present numerical implementation, this library consists of a series of PNG or OBJ files for 2D and 3D patterns. For the 2D cases, the pixel values of the PNG files are directly interpreted as a material density between 0 and 1. For the 3D cases, the surface meshes described by the OBJ files are each voxelized by computing a signed distance field to the surface (Bærentzen and Aanæs, 2002). The distance values can then be interpreted as material densities similarly to the 2D cases. The main benefit of using the present file formats is that it allows easily hand-painting or adjusting the patterns using existing image processing or modeling software. The appearance constraint of the structure generated during optimization is measured relatively to this geometrical pattern library. There is no intrinsic limit to the number $|I|$ of patterns or their spatial resolution, but it should be noted that larger libraries will increase computational run times.

3.2.2 Patch Discretization

A patch ω_i is assigned to each finite element i of the design domain Ω (see figure 3). Patches are rectangular windows of size $|\omega| = (2 \times l_x + 1) \times (2 \times l_y + 1) \times (2 \times l_z + 1)$ where l_x, l_y, l_z are integers defining the half-sizes of the patch along each axis. Thus, ω_i contains all elements in the neighborhood of i satisfying $\omega_i = \{j \mid \|(j_k - i_k)\|_\infty \leq l_k, k \in \{x, y, z\}$. The patch sizes are typically chosen manually or calculated heuristically based on the input pattern sizes. This discretization is used to describe the appearance constraint locally by considering the measure of the appearance for each patch relative to the pattern library.

3.2.3 Distance Metric

The evaluation of the appearance constraint is based upon computing a distance value between any patch from Ω with any patch from I. We define the mapping m as a function $m : \Omega \rightarrow I$ from between all possible patches from Ω and the matching patch from the pattern library I according to the chosen distance metric. Given a patch centered on a given element $i \in \Omega$, we associate the element to the corresponding matching patch in the pattern library by its central element $m(i) \in I$. The notation $\rho_{i,j}$ is used to refer to the density of an element j which is in a neighborhood ω_i of i.

The distance metric used throughout this work (unless specified otherwise) is a density weighted sum of the normalized squared norms and is defined as follows:

$$D(\rho_i, \alpha_{m(i)}) = \rho_i \sum_{j \in \omega_i} \frac{(\rho_{i,j} - \alpha_{m(i),j})^2}{2 \sum_{k \in \omega_i} (0.5 - \alpha_{m(i),k}) + \epsilon} \qquad (2)$$

where α represents the relative densities of the corresponding patch for the predefined geometrical pattern library I, and the small number $\epsilon > 0$ ensures the distance is defined in the unlikely event when the patch from the pattern library I is of uniform 0.5 densities.

We conducted numerical experiments for various distance metrics and found that the present formulation provided the most numerical advantages while remaining mathematically simple and intuitive. The numerator $(\rho_{i,j} - \alpha_{m(i),j})^2$ is a convex function, being zero when the two patches are equal and rapidly growing if the two patches are dissimilar. The denominator $2 \sum_{k \in \omega_i} (0.5 - \alpha_{m(i),k}) + \epsilon$ is a patch-dependent normalization coefficient scaling the impact of each patch as well as widening the range of the appearance constraint values. The weighting factor ρ_i guides the optimization to focus on high-density regions while disregarding void regions. Additionally, it has the benefit of ensuring that the appearance constraint always has a feasible solution.

3.2.4 Brute Force Matching

The mapping function m is stored as an array of dimension $|\Omega|$. One could search exhaustively for the best matching given a distance metric such as Eq. 2. This would be accomplished by evaluating the distance value of each possible pair of patches in the design domain and pattern library, and keeping the matches having the smallest total distance. However, such a brute-force strategy yields a high computation time and will be impractical for any non-trivial scenario. Indeed, an exhaustive search for the 3D optimization problem in Fig. 8(e) requires more than 25×10^9 evaluations of the distance function for each optimization iteration.

As in (Martínez et al., 2015), we will instead apply the fast randomized matching algorithm *PatchMatch* (Barnes et al., 2009) that we extend to handle multiple geometrical patterns. This matching algorithm is described in the following section. The PatchMatch algorithm only provides an approximation of the optimal mapping. Therefore, we compare against the mappings obtained by the brute-force strategy to validate the approximate scheme and adjust its parameters accordingly.

3.2.5 Fast Matching Algorithm

The original PatchMatch algorithm is an iterative process that continually improves the matching assignment. However, in our implementation, we run one execution of the matching process at the start of each topology optimization iteration for evaluating the appearance constraint. Thus, an execution of the matching algorithm aims at both improving the matching assignment and updating the scheme by taking into account the design variables updates from the outer topology optimization loop.

The fast matching algorithm relies on two key assumptions about the characteristic of the present matching problem. Firstly, we assume that a good mapping at a given iteration of the optimization process is likely to be a good starting guess for the mapping at the next topology optimization iteration. Apart from the first couple of topology optimization iterations, where the density field is changing rapidly, we find that this assumption is valid. Secondly, we assume that when a good match in I is found for a given element of Ω then it is likely that the neighboring elements of Ω would also have good matches in the neighborhood of that first match in I. Intuitively, this assumption makes sense since neighboring patches overlap in Ω and therefore, it is to be expected that their optimal matches tend to also overlap I. These two assumptions are often referred to as *time continuity* and *spatial locality*, respectively. Together, they allow to only consider a subset of candidate patches, thus drastically reducing the search space and total number of distance function evaluations. Specifically, when searching the match $m(i)$ of a given element i, using our implementation and choices of parameters, we only consider on average 30 to 50 potential candidate patches of the geometrical pattern library. Numerical experiments show that these settings allow achieving optimized designs qualitatively equivalent to those obtained with an exhaustive search but using a fraction of the computational cost.

The matching assignment is kept in memory after the initialization step and continually updated throughout the entire topology optimization iterations. Furthermore, we extended the matching algorithm to handle multiple patterns. A flowchart of the process is shown in Fig. 2.

Each step of the matching algorithm is described and illustrated in Fig. 3 as well as the gradient computation in the following.

Initialization: Each element $i \in \Omega$ is assigned a match $m(i)$ at random for the beginning of the first topology optimization iteration. It is improbable that a random initialization would, by chance, yield a good matching everywhere for all elements in the design domain. However, this strategy is justified by the fact that it is also equally unlikely that it will yield a bad matching for all elements. Indeed, the subsequent search and propagation steps only need a few good matches to quickly spread and improve the matching for the entire design domain.

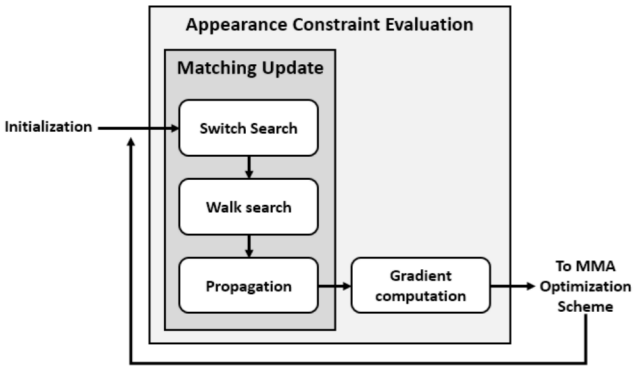

Figure 2: Flowchart of the appearance constraint evaluation including the matching update step for the patterns of the geometrical library.

This initialization can also be done using user-defined patterns for instance by initializing the matching assignment of certain areas of the design domain with elements targeted in specific patterns of the library. Then it is possible to induce a design convergence bias promoting the chosen geometrical pattern for a given area of the design domain.

Switch Search: This phase scans through the pattern library allowing the algorithm to change the pattern for which it determines the nearest patch. We attempt to improve the matching assignment by testing a set of n_S patches for each other pattern of the geometrical library. The selection follows a uniform distribution on the subspace for the maximum dimension of each pattern. Unless specified otherwise, we use $n_S = 5$ for the numerical examples.

Walk Search: We attempt to improve the matching assignment by testing several patches present inside a circle of decreasing radius centered on its current best match. We update the center of the search during the Walk Search if a better match is found to improve the convergence speed as proposed in (Panareda Busto et al., 2010). The coordinates of the tested element are then obtained by $m(i) + (\frac{1}{2})^k w R_k$ where $m(i)$ are the coordinates of the current best matching assignment, $R_k \in [-1, 1] \times [-1, 1] \times [-1, 1]$ is a uniform random vector, w is the maximum search radius being equal to the maximum pattern dimension and clamped to the bounds of the pattern. We examine patches for $k = 0, 1, 2, ...$ until the current search radius $(\frac{1}{2})^k w$ is below 1 voxel.

Propagation: If two elements are next to each other in the design domain there is a high chance that their best matches are also next to each other in the geometrical pattern library. Therefore, we improve the matching assignment by using a propagation algorithm similar to the *Distance Transform* algorithm (Kimmel et al., 1996) (Felzenszwalb and Huttenlocher, 2004). The propagation is performed in a forward pass followed by a backward pass using the connectivity information of the finite element mesh to ensure a coherent spreading. The propagation algorithm sweeps through the design domain and visits each element sequentially. In 3D, each element i has at most 26 neighboring elements, 13 of which have not yet been visited by the current pass. For these 13 neighbors, a shifted version of the matching $m(i)$ is considered as a candidate for a better match. Computing this propagation sequentially allows potentially spreading the matching over the entire design domain in only two passes. This means that the algorithm performance scales linearly with the number of elements.

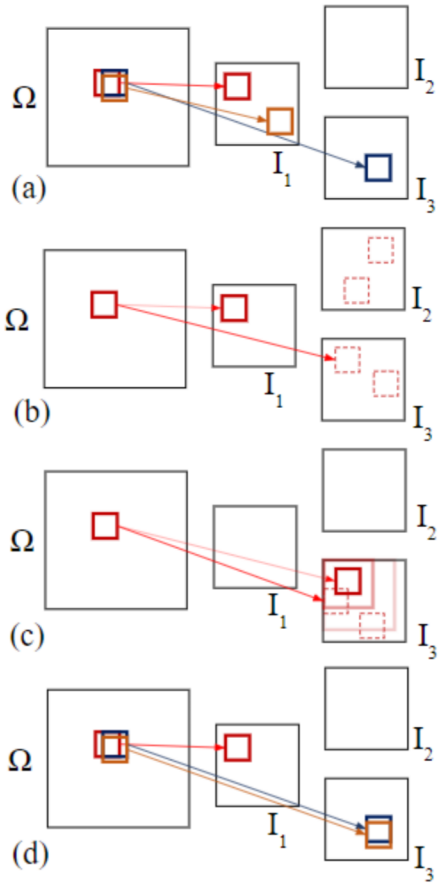

Figure 3: Illustration of the fast randomized matching algorithm. (a) Current best matches are initialized at random in the *Initialization* phase. (b) Each patch randomly searches for a better match in the other input patterns at the *Switch Search* phase. (c) Each patch randomly searches for a better match inside progressively tighter neighborhoods around its current best match at the *Walk Search* phase. (d) Each patch attempts to propagate a shifted version of its current best match to its neighbors at the *Propagation* phase.

CPU Parallelization: Our implementation of the matching algorithm is adapted to multi-threading for computational performance. This is an important factor as the matching algorithm is computationally expensive for the topology optimization of high resolution 3D density fields. CPU parallelization

is trivial to implement for all phases of the matching process with the exception of the propagation being inherently a serial operation. Thus, we split the design domain and run the propagation algorithm on one thread for each contiguous sub-space. As the propagation operates in two passes then this domain splitting does not prevent spreading matches through the borders of the sub-domains. In theory, the quality of the matching obtained using the parallelized implementation of the propagation algorithm should be inferior. However, in practice, we find that the total distance of the obtained matching is essentially unaffected for our numerical experiments using 8 threads and 8 sub-domains.

Randomness: The fast matching algorithm is an iterative scheme approximating the optimal matching and therefore, involves randomness. Moreover, the matching process is itself extremely non-convex. Therefore, any topology optimization using an appearance constraint is likely to converge to a local minimum. Experimentally, we show in a later section that although the designs might differ in each local minimum, their optimized compliance values are practically equivalent. Moreover, the pseudo-random number generators applied within the evaluation of the appearance constraint can be deterministically seeded in order to control the randomness of the matching assignment and obtain easily repeatable numerical results.

Low Density Areas: The matching has no impact in a region where material is void due to the ρ_i weight in the distance metric in Eq. 2. Thus, it is unnecessary to search for a patch assignment in void regions. In our implementation, all patches having an average density $\frac{1}{|\omega_i|}\sum_{j\in\omega_i}\rho_j$ below a threshold value of 0.05 are excluded from the matching search phases. This scheme has proven effective at reducing the search space for large resolutions and for small and moderate volume fractions, where larger areas of voids are present in the optimized designs.

Boundary Handling: The distance between two patches is not well-defined if one of the patches is partially outside of the design domain Ω or one of the patterns in I. We assume that regions outside of the design domain are void and therefore, assume their density to be zero. This zero-padding strategy ensures that all elements of Ω have well defined gradients. However, using the same strategy for patches partially lying outside of I leads to adverse effects and fictitious geometrical features appearing in the optimized design. Therefore, only patches lying entirely inside patterns of the library are considered as candidates for the matching algorithm.

3.2.6 Appearance Function for constraint

Having calculated the mapping function $m : \Omega \rightarrow I$ using the fast matching algorithm, we now formulate the appearance function for the constraint. It is constructed as an aggregation across all elements for the design domain of the distances $D(\rho_i, \alpha_{m(i)})$ defined in Eq. 2. Using a simple sum aggregation ensures well defined first order derivatives and yields the following appearance constraint:

$$A(\rho) = \frac{1}{A^*|\Omega|}\sum_{i\in\Omega} D(\rho_i, \alpha_{m(i)}) \leq 1 \tag{3}$$

The coefficient $A^* \in]0, 1]$ is the value of the appearance constraint, analogous to G^* for the volume fraction constraint in Eq. 1. The summed appearance in Eq. 3 is normalized by the number of elements considered in Ω. It is possible to use other aggregation functions for the appearance constraint to obtain different mathematical behaviors. For instance, the effect of using a soft-min aggregation function is investigated in Section 5.

The derivatives of the appearance function with respect to the design variables are computed using Eq.2 and Eq.3 as follows:

$$\frac{\partial A}{\partial \rho_j} = \frac{1}{A^*|\Omega|}\sum_{i|j\in\omega_i} \begin{cases} \text{If } i = j: \\ \frac{D(\rho_i,\rho_{m(i)})}{\rho_i} + \rho_i \frac{(\rho_j - \alpha_{i,j})}{\sum_{j\in\omega_i}(0.5-\alpha_{i,j})^2+\epsilon} \\ \\ \text{If } i \neq j: \\ \rho_i \frac{(\rho_j - \alpha_{i,j})}{\sum_{j\in\omega_i}(0.5-\alpha_{i,j})^2+\epsilon} \end{cases} \tag{4}$$

The calculated gradient are then implemented in the gradient-based optimization scheme as shown by the flowchart in Fig. 2.

As the Fast Matching algorithm is a discontinuous operation primarily due to the switch Search and Walk Search steps, the appearance function is non-smooth and only piece-wise differentiable. Nevertheless, our numerical experiments show that this expression can be used stably in our numerical framework mostly thanks to the fast convergence of the matching algorithm. This is further discussed in section 4.2.

4 Results and Analysis

The previous section described a numerical optimization implementation for generating optimized structures having enforced geometrical appearances. The present section uses this approach in various scenarios and highlights its advantages and shortcomings. Initially, we discuss a methodology for choosing the threshold value A^* controlling the appearance constraint. Afterwards, we evaluate the behavior of the optimization scheme through several 2D and 3D numerical experiments for which we evaluate computational run time and optimization convergence history.

4.1 Choosing Appearance constraint value A^*

In the following numerical experiment, we study the optimized designs and their compliance value obtained using five different appearance constraint values as $A^* = \{0.01, 0.02, 0.04, 0.08, 0.16\}$. A fixed number of 100 optimization iterations is chosen as the stopping criterion and all

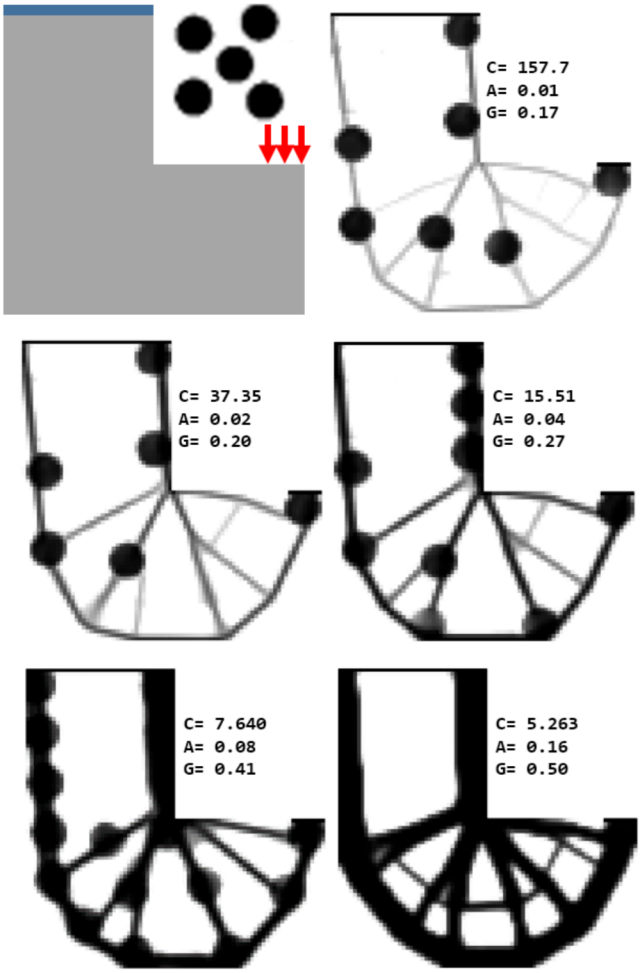

Figure 4: Comparison of different constraint values A^* on a 100×100 2D L-shape mesh. The design specifications are shown in the top left with boundary conditions and loads in blue and red, respectively. The pattern is an image of 5 disconnected dots. Five optimized designs and their compliance (C), appearance (A), and volume fraction (G).

models have the same volume fraction constraint of $G^* = 0.5$. The optimized results are shown in Fig. 4.

It is clear for this optimization setup that a geometrical pattern consisting of disconnected circular dots will not be optimal for minimized structural compliance as such a structure will typically consist of bars. Nevertheless, it is promising to see how the optimization formulation behaves for such a guided geometrical pattern. Firstly, we observe that all constraints are satisfied in all cases and the optimized designs exhibit the guided geometrical shape. Secondly, we see that tighter appearance constraint values A^* lead to designs having easily recognizable patterns in the resulting optimized design, but this also causes a significant increase in compliance. Thirdly, we observe that in order to satisfy the appearance constraint, the element densities for the optimized design are reduced outside of the circular dots. This justifies why the volume fraction constraint is sometimes inactive. Other numerical experiments yield similar findings and show that the topology optimization process is quite sensitive to the choice of A^*.

In (Martínez et al., 2015) the problem formulation consists of an objective minimizing the appearance subject to a constraint on compliance, for which the user chooses a threshold. The volume constraint and the redistribution of material for the close to void regions commonly prevent the emergence of disconnected components when the compliance constraint is active. However, having a high compliance threshold and a geometrical pattern having disconnected components such as the one shown in Fig. 4, results in designs that could have disconnected components. A self-weight loading scenario for such a case will eliminate this issue. However, the self-weight loading scenario is unnecessary in our numerical framework because the compliance is minimized as an objective function. Nevertheless, the user has to choose a threshold A^* for the appearance constraint in our problem formulation. A poor choice of threshold value can lead to designs exhibiting non-converged densities or unrecognizable geometrical patterns. Therefore, we propose a heuristic method to choose the A^* constraint value.

The challenge concerning the appearance constraint lies in the fact that it has no explicit physical meaning and instead relies on a more abstract and subjective idea of aesthetics and visual similarity between the optimized design and the geometrical patterns of a library. In contrast, it is easy to choose a G^* value for the volume fraction constraint. Applying the appearance constraint requires an additional effort in adjusting A^* to determine the value best suited for a given problem. We suggest a heuristic method to determine a suitable value for A^* requiring no trial and error or additional optimization iterations. Run the topology optimization for N iterations without constraining the appearance. Then, set the appearance constraint value A^* to a fraction γ of the current appearance score $A_N(\rho)$ and resume the topology optimization at iteration $N+1$ as follows:

$$A^* = \gamma A_N(\rho) \tag{5}$$

Unless stated otherwise, we use this approach with $N = 10$ and $\gamma = 0.70$ in all subsequent examples.

4.2 Optimization convergence behavior

The following numerical experiment aims at evaluating the convergence behavior of the optimization using an appearance constraint. For this purpose, we use a cantilever beam scenario on a 384×256 grid resolution subject to a volume fraction constraint $G^* = 0.4$. Comparative tests are done using different geometrical pattern libraries including oriented pattern(s) or a single detailed high-resolution pattern. The choice for the size of both the patterns provided to the geometrical library and the patches used for the evaluation of the appearance constraint depends on the resolution of the mesh and the detail level for the geometrical features of the patterns. Indeed, patterns with detailed geometrical features require larger patches to ensure propagation while smaller patches are applied in the

case of repeated patterns. Finally, the value of the threshold A^* is chosen using the methodology described in Eq. 5. The optimized structures obtained for different pattern libraries are shown in Fig. 5.

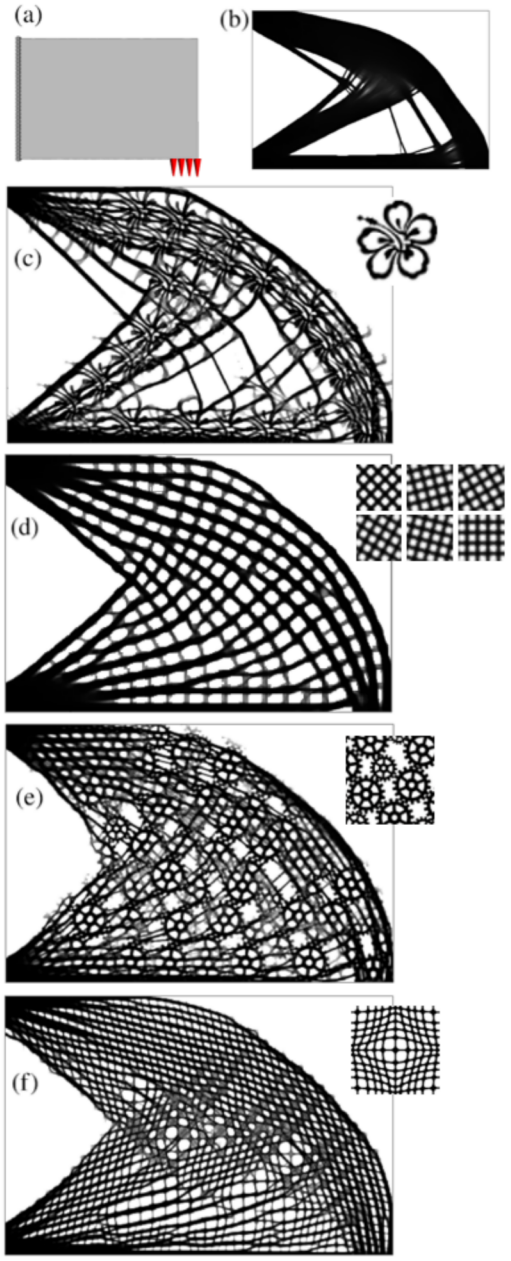

Figure 5: (a) 2D cantilever fixed on the left side and loaded at the bottom right. (b) Optimized design without appearance constraint as a baseline. (c), (d), (e), (f) Optimized designs using the appearance constraint for geometrical pattern libraries consisting of one or multiple patterns shown at the top right corner of each design.

The settings chosen for each optimization run are summarized in the following table:

Case	Pattern	Patch	A^*	It. runtime	Compl.
5(b)	-	-	-	5.24s	4.3×10^6
5(c)	80×80	31×31	0.18	13.35s	4.7×10^6
5(d)	75×75	21×21	0.18	14.82s	5.2×10^6
5(e)	80×90	31×31	0.18	15.01s	5.3×10^6
5(f)	90×90	21×21	0.20	10.56s	6.1×10^6

The convergence history of the five cantilever designs is shown in Fig. 6 for the compliance value, volume fraction, and appearance constraint. Note that since the appearance constraint is only applied after $N = 10$ iterations, the history plot starts at iteration N.

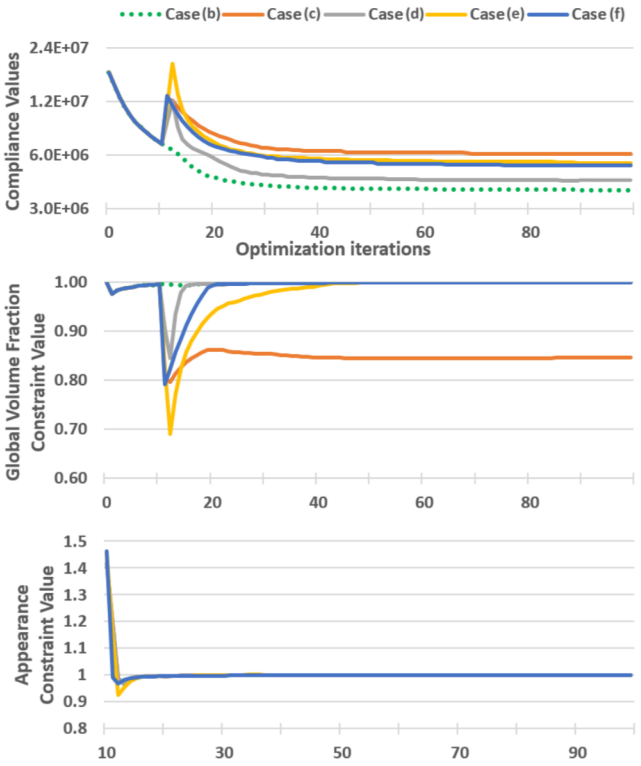

Figure 6: Convergence history for the five designs in Fig. 5. Case (b) shows the design without appearance constraint in Fig. 5(b). Cases (c), (d), (e) and (f) correspond to the designs in Fig. 5(c), 5(d), 5(e) and 5(f) respectively. The appearance constraint is disabled for the first 10 iterations, hence the convergence history starts at iteration 10.

The compliance history shows a smooth decrease across all optimization iterations except for the peak at iteration $N = 10$ caused by the introduction of the appearance constraint. The initial violation of the appearance constraint value incurs a rapid change in the material distribution. With the exception of iteration N, both the appearance constraint and the volume fraction constraint are satisfied and stable during the optimization iterations.

Naturally, Case (b) (Fig. 5(b)), having no appearance constraint achieves the lowest compliance. Case (d) (Fig. 5(d)) achieves the second best compliance due to its pattern library consisting of six angled grid-like patterns, analogous

to oriented orthotropic microstructures. Note, that this design is similar results obtained with the so-called Local Volume constraint (Wu et al., 2016) as well as recent work tackling de-homogenization and microstructure generation reviewed in (Wu et al., 2021). Case (f) (Fig. 5(f)) exhibits a similar result but its geometrical features are significantly thinner and extracted from small patches inside the distorted grid pattern present in the library. Finally, Case (c) (Fig. 5(c)) and Case (e) (Fig. 5(e)) are subjected to an appearance constraint having larger and intricate geometrical patterns leading to a more artistic result. Case (c) (Fig. 5(c)) is an example where the geometrical pattern is mostly void and the optimization reduces the density values in order to satisfy the appearance constraint. Therefore, the volume fraction constraint from Case (c) is inactive and 15% lower than the constraint value.

Our numerical experiments on several other test scenarios also show that both the patch size and the value of A^* influence the compliance optimization history. Indeed, larger patches constrain the guided appearance of the generated structure by penalizing the non-propagation of patterns over larger regions. However, it is not possible to use smaller patches for patterns having detailed geometrical features like the ones of case (c) and (e) as these are unable to include all their detailed geometrical features and thereby, fail to reproduce these geometrical features correctly.

The update of the mapping $m : \Omega \rightarrow I$ via the Fast Matching algorithm is a discontinuous operation. Nevertheless, the convergence history for all our numerical experiments is smooth and stable. We attribute this observation to three key factors. Firstly, the Fast Matching algorithm converges extremely quickly. Empirically, it achieves an approximate mapping having a total distance within 1% of the true optimal mapping in generally less than 5 iterations. Secondly, the patches overlap in the design domain having the effect of averaging the influence of each individual patch. Thirdly, the mapping is only updated when a better match is found for a given patch in the library. Therefore, the design updates and the mapping updates are actually cooperating throughout the optimization process.

4.3 Non-Convexity

Even for a simple density-based compliance topology optimization problem subject to a volume faction constraint the solution space is known to be non-convex. Despite the absence of theoretical guarantees, the general consensus is that initializing all relative densities to intermediate values combined with length scale control and move limits is usually enough to prevent the optimization to end up in a poor local minimum.

As discussed in Section 3, the appearance constraint significantly increases the non-linearity and non-convexity of the solution space. The Fast Matching algorithm introduces randomness in the initialization, switch search and walk search phases. Thus, running the same optimization scenario using different random seeds usually results in converging toward a different local minimum. We empirically conclude, by optimizing a large number designs, that the optimized topologies differ, but their mechanical performance is equivalent. Fig. 7 shows the compliance and designs obtained for four optimization runs of the cantilever problem shown in Fig. 5(a) using the same parameters but different random seeds. Note, we do not need to run N preliminary iterations without appearance constraint to choose an appropriate value for A^* since we use the same predetermined value of $A^* = 0.15$ for every run of this experiment.

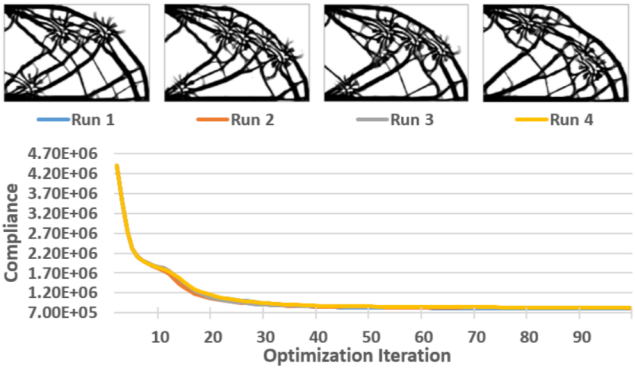

Figure 7: Comparison of the compliance optimization history for four designs using the setup at Fig 5(a) and the geometrical pattern library at Fig. 5(c). The converged designs from run 1 to 4 have a final compliance of 8.07×10^5, 8.19×10^5, 8.13×10^5 and 8.19×10^5, respectively.

4.4 Numerical performance

The numerical implementation was done in C++. The wall-clock execution time reported in table for Fig. 5 is measured using a laptop having all calculations running on its six cores 2.60 GHz Intel Core i7 - 9850H CPU. The Jacobi preconditioned gradient descent solver as well as the appearance constraint are multi-threaded as discussed in Section 3. For such optimization problems, the matching and appearance constraint computation is roughly half of the total optimization time. Computation time differences between each design optimization are explained by the patch sizes and the number of geometrical patterns in the libraries.

4.5 3D designs

We apply the appearance constraint for a 3D cubic design domain having resolution of 102 x 102 x 102 yielding approximately 1×10^6 design variables and 3×10^6 degrees of freedom. Loading is applied uniformly on the top of the design domain clamped at the four bottom corners as shown in Fig. 8(a). The volume fraction constraint is set to $G^* = 0.5$ and the value of A^* is set using the methodology in Eq. 5. The baseline model without appearance constraint is shown in Fig. 8(a). The four optimized designs subject to the appearance constraint and the associated geometrical patterns illustrated in blue are shown in Fig. 8(b), (c), (d) and (e).

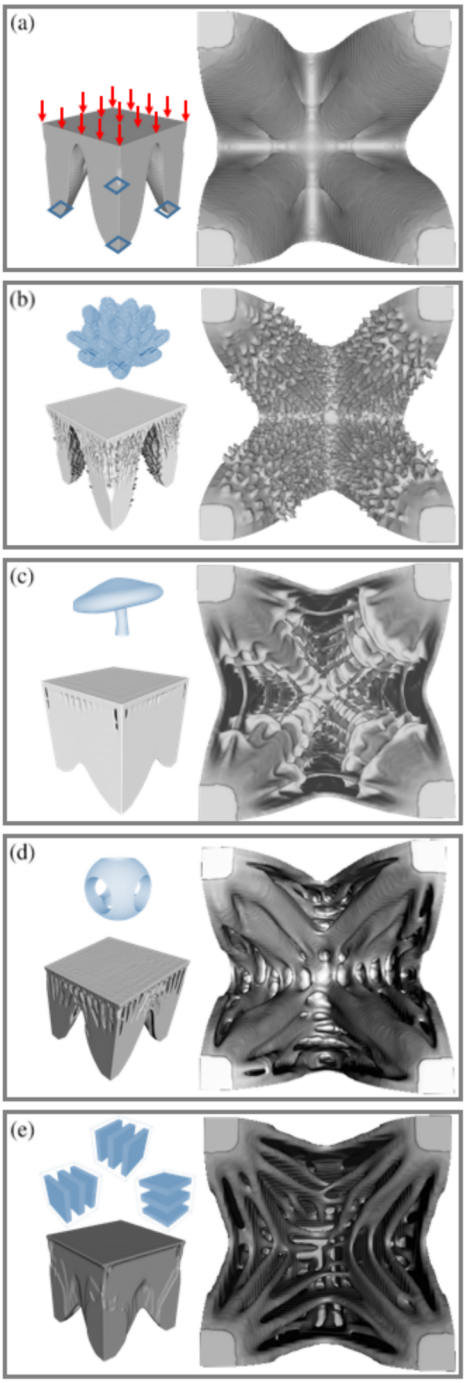

Figure 8: 3D designs obtained using the appearance constraint for various geometrical libraries. The model (a) without appearance constraint yielding a compliance of 1.0 as baseline. The models (b), (c), (d) and (e) have optimized compliances of 1.25, 1.24, 1.49, and 1.13 respectively.

We find that the appearance constraint is satisfied and the objective function decreases smoothly during the optimization for the 3D experiments, similarly to the previously shown 2D examples. The baseline model 8(a) also achieves the lowest compliance since it is not subject to the additional appearance constraint. The appearance constraint thus succeeds at changing the geometrical appearance of the optimized designs to partially resemble the geometrical patterns of the various libraries.

Designs 8(b) and (c) have a target geometrical pattern for representing a lotus flower and a mushroom, respectively. The two optimized designs mainly exhibit aesthetic changes near their surfaces, where outward spikes or flat horizontal planes appear. For design 8(d), the target geometrical pattern is a sphere hollowed by three axis-aligned cylinders. The optimized design consists of tubular membranes which split and connect at regular intervals from the bottom to the top plate of the design domain. Finally, design 8(e) has a pattern library consisting of three different parallel plates aligned along the X, Y and Z directions, respectively. The optimized design consists of a selection of axis-aligned membranes. Note, that only two of the three input patterns present in the library are to be found in the optimized design. Indeed, the pattern having plates orthogonal to the load direction is of no use when minimizing compliance and is quickly removed by the concurrent optimization for the geometrical pattern mapping and design variables.

5 Extensions

In this section, various modifications and extensions for the proposed appearance constraint are explored. Specifically, we change the objective function of the topology optimization formulation, the design variable type, and the aggregation function applied for the appearance constraint.

5.1 Stress minimization

The appearance constraint is dependent on the density field but independent from the objective function. Here, we consider stress minimization as objective instead of minimizing compliance. A p-norm aggregation of the von Mises stress with q-relaxation is applied using $n = 10$ and $q = 0.5$ (Duysinx and Bendsøe, 1998) (Duysinx and Sigmund, 1998) (Lee et al., 2011). The results for the design of an L-Shape model when minimizing compliance versus minimizing peak stress are shown in Fig. 9c and Fig. 9d, respectively.

Both designs 9(c) and 9(d) have the same geometrical pattern and the same value $A^* = 0.16$. The design 9(d) successfully avoids the stress singularity near the concave corner of the design domain. This optimization result shows that the appearance constraint is compatible with other objectives in the topology optimization formulation.

5.2 Material Orientation as Design Variables

The approach can also be extended to design variables other than element density. One such design variable is the local orientation of anisotropic materials. For the present optimization formulation each element is associated with a density

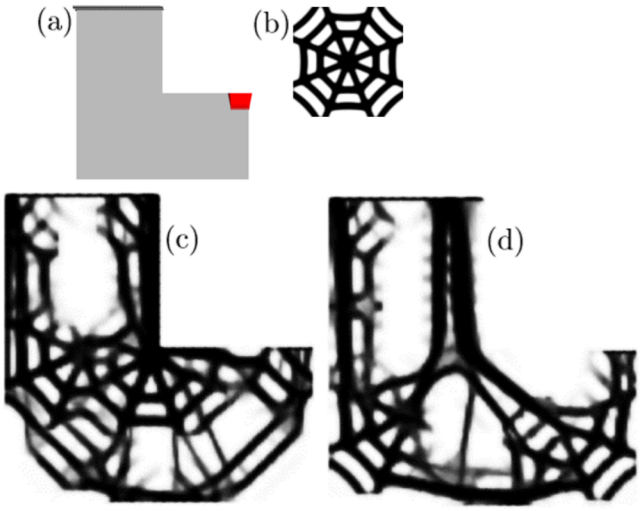

Figure 9: (a) L-Shape model. (b) Geometrical pattern for the appearance constraint library. (c) Design minimizing compliance. (d) Design minimizing peak stress. Design (d) has a 27% increase in compliance and a 66% decrease in relaxed stress compared to design (c).

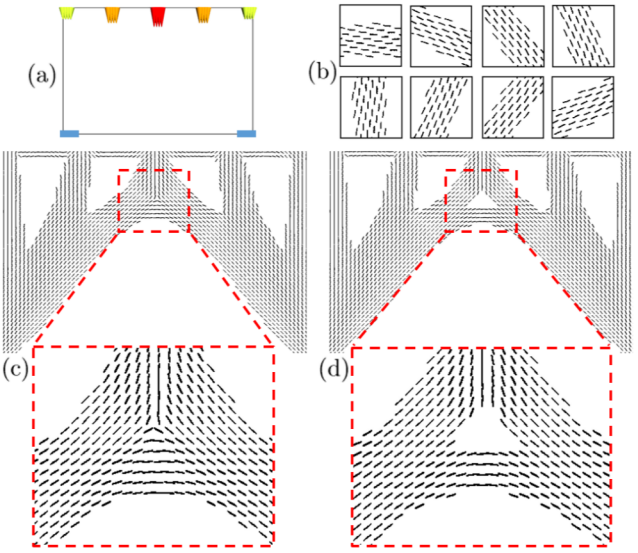

Figure 10: (a) Bridge design model having five loads at the top and fixed boundary conditions at the bottom corners. (b) Geometrical pattern library for the appearance constraint where local material orientations are visualized as collections of short line segments. (c) Optimized bridge model yielding an orientation singularity before applying the appearance constraint. (d) Optimized bridge model after applying the appearance constraint resolving the singularity in the optimized angle design variables. The models (c) and (d) have a final compliance of 3.57×10^4 and 3.60×10^4.

design variable as well as a material angle design variable defining the orientation of the orthotropic material (Schmidt et al., 2020). Both material density and orientation are optimized using a gradient-based scheme to minimize compliance subject to a volume fraction constraint. Two implementation modifications are introduced due to the appearance constraint. Firstly, the geometrical patterns of the library now contain an angle value in addition to the density value at each point. Secondly, the distance metric computes the difference between two angles including the π-symmetry of the orientation design variables as follows:

$$D(\rho_i, \theta_i, \psi_{m(i)}) = \rho_i \sum_{j \in \omega_i} \frac{min(2\pi - d, d)^2}{|\omega_i| \times \pi/2} \quad \text{with} \quad d = abs(\theta_{m(i),j} - \psi_{m(i),j}) \tag{6}$$

where θ_i are design variables for the material orientations and $\psi_{m(i)}$ are the associated patches in the pattern library, analogous to ρ_i and $\alpha_{m(i)}$ for the material densities. The optimization result of this formulation is shown in Fig. 10 comparing the results for a bridge model.

The model 10(c) optimized having no appearance constraint yields a material orientation singularity at the center of the design domain due to the multi-axial stress at this particular region. Providing a geometrical pattern library 10(b) without singularities enables the appearance constraint to enforce a design where the angle design variable singularity is resolved. Indeed, we observe that every patch of the optimized design 10(d) can be mapped to a patch inside one of the eight patterns of the library.

5.3 Soft-min Aggregation

It is possible to induce different behaviors on the composition of the geometric features in the generated design by modifying the aggregation function in $A(\Omega)$. A soft-min aggregation function pushes the optimization to only reproduce the pattern in localized regions where the appearance value is lowest. Conceptually, this means that the soft-min constraint formulation yields a design where the pattern should appear only in a single region having a high fidelity. In contrast, the previously applied sum aggregation formulation yields designs where the patterns appear everywhere in the domain with lower fidelity.

To prevent the optimizer from "choosing" an empty area as the region having minimum distance, we use a simple quadratic form for the distance metric without multiplying by the density of the central element ρ_i:

$$D(\rho_i, \alpha_{m(i)}) = \sum_{j \in \omega_i} \frac{(\rho_{i,j} - \alpha_{m(i),j})^2}{2 \sum_{k \in \omega_i} (0.5 - \alpha_{m(i),k}) + \epsilon} \tag{7}$$

The distance function $D(\rho_i, \alpha_{m(i)})$ is shortened to $D(\rho_i)$ in the following equations for conciseness. Therefore, we have the following formulation of the aggregated appearance function using a soft-min expression

$$A(\rho) = \frac{1}{A^*} \frac{\sum_{i \in \Omega} D(\rho_i) e^{\eta D(\rho_i)}}{\sum_{i \in \Omega} e^{\eta D(\rho_i)}} \tag{8}$$

where we use $\eta = -8$ as value for the exponent controlling how accurate the soft-min expression matches a true min function. The gradient of this formulation is:

$$\frac{\partial A}{\partial \rho_j} = \frac{1}{A^* \sum_{k\in\Omega} e^{\eta D(\rho_k)}} \sum_{i|j\in\omega_i} e^{\eta D(\rho_i)} [1 + \eta(D(\rho_i) - \frac{\sum_{k\in\Omega} D(\rho_k) e^{\eta D(\rho_k)}}{\sum_{k\in\Omega} e^{\eta D(\rho_k)}})] \frac{(\rho_{i,j} - \alpha_{m(i),j})}{\sum_{l\in\omega_i}(0.5 - \alpha_{m(i),l}) + \epsilon} \tag{9}$$

Fig. 11 shows the generated design on a 160×80 mesh resolution using the parameters and geometrical pattern library from Fig. 5(f).

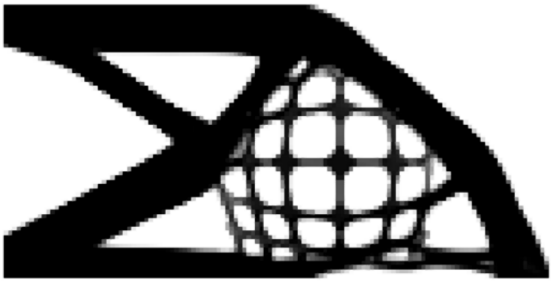

Figure 11: Cantilever design scenario from Fig. 5(a) obtained using a soft-min appearance aggregation function and the geometrical pattern of Fig. 5(f).

We observe that the provided geometric pattern is reproduced only once in the design without being present anywhere else. An interesting application of the soft-min aggregation function is therefore "branding", consisting in enforcing a logo provided as a pattern to appear once in the generated structure at a minimal stiffness cost. Our experiments reveal that this formulation of the appearance constraint tends to have a significantly smaller impact on the final value of the objective function compared to the sum aggregation. Moreover, we note that it usually takes only one instance of the pattern to appear in the design for it to be easily recognizable at a glance. Fig. 12 illustrates this "branding" application using the soft-min aggregation function to incorporate a logo into an optimized design. Additionally, one should note that each pattern and logo appears in a different location of the design domain in Fig. 11, 12(a) and 12(b). This is due to each pattern having distinct mechanical properties. Thus, the optimization scheme aims to find the best location to faithfully reproduce each pattern at a minimal cost in mass and stiffness.

6 Conclusion

We propose a formulation for style transfer in density-based topology optimization using a geometrical pattern library. Extending the pioneering work from (Martínez et al., 2015), the approach applies a fast matching algorithm to create a mapping from regions of the design domain to patches of the

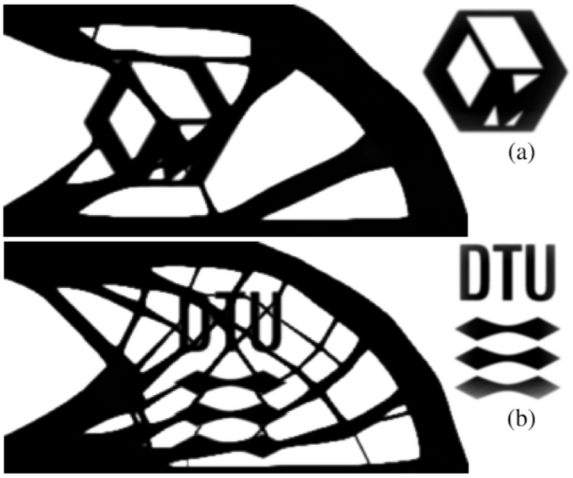

Figure 12: Cantilever design scenario from Fig. 5(a) optimized on a 300×150 mesh obtained using a soft-min appearance aggregation function and a pattern library made of the logos from the Laboratory of Mathematics at INSA (LMI) (a), and the Technical University of Denmark (DTU) (b), respectively.

geometrical patterns present in the library. A constraint controls the appearance of the geometrical patterns in the design and is integrated into a gradient-based optimization loop. The appearance constraint is demonstrated for a variety of 2D and 3D compliance minimization scenarios. Additionally, we demonstrate the application of the appearance constraint for different objectives by formulating a stress minimization. The optimization formulation is also extended to address material orientations as design variables. Here the appearance constraint can prove effective at steering the optimization result away from pathological singularities in the angles and a design consisting of a discrete set of anisotropic materials.

Future Work

The proposed formulation is invariant to translation of the geometrical patterns but not to rotation or scaling. To allow for patterns to appear with different orientations or scale then one would need to pre-transform the geometrical pattern library for rotated or scaled variations of the patterns. The drawback is a significant increase in the computational cost for the matching process. An alternative approach can be to employ descriptor vectors commonly used in the field of Computer Vision since those would encapsulate translation, rotation and scaling invariance at no additional computational cost when evaluating the distance metric. However, the matching propagation would be challenging to implement. Yet another strategy inspired by the field of Machine Learning is to use the latent space of convolutional neural networks or auto-encoder networks to distill geometrical features of patterns into compact descriptor vectors. Preliminary work on this idea has been performed in (Vulimiri et al., 2021).

Replication of Results

Unless explicitly stated in the text, all results presented in this work apply the same default parameters: $p = 3$, $Y_{min} = 10^{-6}$, $Y_0 = 1$. We use a density filter of radius 1.5 relative to the element size in regular 2D or 3D grid meshes consisting of squares or cubes, respectively. The patch dimensions are $l_x = 8$, $l_y = 8$ and $l_z = 8$ in 3D cases and $l_x = 0$, $l_y = 8$ and $l_z = 8$ in 2D cases. All results are obtained in our C++ Topology Optimization prototyping framework with a dependency on the open-source Eigen library for solving linear systems of equations, although any other similar linear algebra library would be equally suitable.

Acknowledgments

Ole Sigmund would like to acknowledge the support of the Villum Foundation, Denmark through the Villum Investigator Project InnoTop. All authors acknowledge the reviewers for their comments helping to improve the manuscript.

Conflict of Interest Statement

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. (2009). PatchMatch: a randomized correspondence algorithm for structural image editing. *SIGGRAPH '09*.

Barnes, C. and Zhang, F.-L. (2016). A survey of the state-of-the-art in patch-based synthesis. *Computational Visual Media*, 3.

Bendsøe, M. P. and Sigmund, O. (1999). Material interpolation schemes in topology optimization. *Archive of Applied Mechanics*, 69(9):635–654.

Bendsøe, M. P. and Sigmund, O. (2003). *Topology Optimization - Theory, Methods, and Applications*. Springer Verlag, Germany.

Bærentzen, A. and Aanæs, H. (2002). Generating Signed Distance Fields From Triangle Meshes. *IMM Technical Report*.

Chen, W., Xia, X., Xin, S., XIA, Y., Lefebvre, S., and Wang, W. (2016). Synthesis of Filigrees for Digital Fabrication. *ACM Transactions on Graphics*, 35(4). Publisher: Association for Computing Machinery.

Deng, H., Vulimiri, P. S., and To, A. C. (2021). An efficient 146-line 3D sensitivity analysis code of stress-based topology optimization written in MATLAB. *Optimization and Engineering*.

Dumas, J., Lu, A., Lefebvre, S., Wu, J., and Dick, C. (2015). By-Example Synthesis of Structurally Sound Patterns. *ACM Transactions on Graphics*, 34.

Duysinx, P. and Bendsøe, M. P. (1998). Topology optimization of continuum structures with local stress constraints. *International Journal for Numerical Methods in Engineering*, 43(8):1453–1478.

Duysinx, P. and Sigmund, O. (1998). New developments in handling stress constraints in optimal material distribution. In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. American Institute of Aeronautics and Astronautics.

Díaz, A. and Sigmund, O. (1995). Checkerboard patterns in layout optimization. *Structural optimization*, 10(1):40–45.

Felzenszwalb, P. and Huttenlocher, D. (2004). Distance Transforms of Sampled Functions. *Theory of Computing*, 8.

Gao, X. and Ma, H. (2015). Topology optimization of continuum structures under buckling constraints. *Computers & Structures*, 157:142–152.

Gumin, M. (2016). Procedural generation from a single example with Wave Function Collapse Algorithm.

Hoffarth, M., Gerzen, N., and Pedersen, C. (2017). ALM Overhang Constraint in Topology Optimization for Industrial Applications. In *12th World Congress on Structural and Multidisciplinary Optimisation*.

Kimmel, R., Kiryati, N., and Bruckstein, A. M. (1996). Subpixel distance maps and weighted distance transforms. *Journal of Mathematical Imaging and Vision*, 6(2):223–233.

Kwatra, V., Essa, I., Bobick, A., and Kwatra, N. (2005). Texture optimization for example-based synthesis. *ACM Trans. Graph.*, 24:795–802.

Langelaar, M. (2016). Topology optimization of 3D self-supporting structures for additive manufacturing. *Additive Manufacturing*, 12:60–70.

Langelaar, M. (2017). An additive manufacturing filter for topology optimization of print-ready designs. *Structural and Multidisciplinary Optimization*, 55(3):871–883.

Lee, E., James, K., and Martins, J. (2011). Stress-Constrained Topology Optimization with Design-Dependent Loading. *Structural and Multidisciplinary Optimization*.

Martínez, J., Dumas, J., Lefebvre, S., and Wei, L.-Y. (2015). Structure and appearance optimization for controllable shape design. *ACM Transactions on Graphics*, 34(6):12. Publisher: Association for Computing Machinery.

Panareda Busto, P., Eisenacher, C., Lefebvre, S., and Stamminger, M. (2010). Instant Texture Synthesis by Numbers. In *Proceedings of the Vision, Modeling, and Visualization Workshop 2010*, volume 2010, pages 81–85.

Schmidt, M.-P., Couret, L., Gout, C., and Pedersen, C. B. W. (2020). Structural topology optimization with smoothly varying fiber orientations. *Structural and Multidisciplinary Optimization*, 62(6):3105–3126.

Sigmund, O. (1997). On the Design of Compliant Mechanisms Using Topology Optimization. *Mechanics of Structures and Machines*, 25(4):493–524. Publisher: Taylor & Francis.

Stava, O., Vanek, J., Benes, B., Carr, N., and Mech, R. (2012). Stress Relief: Improving Structural Strength of 3D Printable Objects. *ACM Trans. Graph.*, 31:48:1–48:11.

Svanberg, K. (1987). The method of moving asymptotes - a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373.

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):4551. Number: 16 Publisher: Multidisciplinary Digital Publishing Institute.

Wei, L.-Y., Lefebvre, S., Kwatra, V., and Turk, G. (2009). State of the Art in Example-based Texture Synthesis. *Eurographics 2009 - State of the Art Reports*. Publisher: The Eurographics Association.

Wu, J., Aage, N., Westermann, R., and Sigmund, O. (2016). Infill Optimization for Additive Manufacturing - Approaching Bone-like Porous Structures. *IEEE Transactions on Visualization and Computer Graphics*, 24:1127 – 1140.

Wu, J., Sigmund, O., and Groen, J. P. (2021). Topology Optimization of Multi-scale Structures: A Review. *Structural and Multidisciplinary Optimization*, 63:1455–1480.