DTU Library

Combined shape and topology optimization of 3D structures

Christiansen, Asger Nyman; Bærentzen, Jakob Andreas; Nobel-Jørgensen, Morten; Aage, Niels; Sigmund, Ole

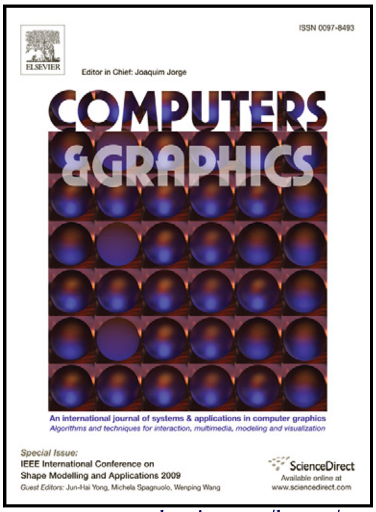
Published in:
Computers & Graphics

Link to article, DOI:
10.1016/j.cag.2014.09.021

Publication date:
2015

Document Version
Peer reviewed version

Author's Accepted Manuscript



www.elsevier.com/locate/cag

Combined shape and topology optimization of 3D structures

Asger N. Christiansen, J. Andreas Bærentzen, Morten Nobel-Jørgensen, Niels Aage, Ole Sigmund

PII: S0097-8493(14)00109-5
DOI: http://dx.doi.org/10.1016/j.cag.2014.09.021
Reference: CAG2511

To appear in: *Computers & Graphics*

Received date: 29 June 2014
Revised date: 24 August 2014
Accepted date: 16 September 2014

Combined Shape and Topology Optimization of 3D Structures

Asger N. Christiansen, J. Andreas Bærentzen, Morten Nobel-Jørgensen, Niels Aage, Ole Sigmund

Technical University of Denmark, Denmark

Abstract

We present a method for automatic generation of 3D models based on shape and topology optimization. The optimization procedure, or model generation process, is initialized by a set of boundary conditions, an objective function, constraints and an initial structure. Using this input, the method will automatically deform and change the topology of the initial structure such that the objective function is optimized subject to the specified constraints and boundary conditions. For example, this tool can be used to improve the stiffness of a structure before printing, reduce the amount of material needed to construct a bridge, or to design functional chairs, tables, etc. which at the same time are visually pleasing.

The structure is represented explicitly by a simplicial complex and deformed by moving surface vertices and relabeling tetrahedra. To ensure a well-formed tetrahedral mesh during these deformations, the Deformable Simplicial Complex method is used. The deformations are based on optimizing the objective, which in this paper will be maximizing stiffness. Furthermore, the optimization procedure will be subject to constraints such as a limit on the amount of material and the difference from the original shape.

Keywords: Topology optimization, shape optimization, Deformable Simplicial Complex method, structural design

1. Introduction

Topology optimization is the discipline of finding the optimal shape and topology of a structure [1][2]. It can be used to solve a wide variety of design problems arising when producing such diverse products as cars, houses, computer chips and antennas. The manufacturers are often concerned with finding the stiffest structure, the lightest structure which does not break, the structure with the highest cooling effect, or the structure with the best flow or highest efficiency.

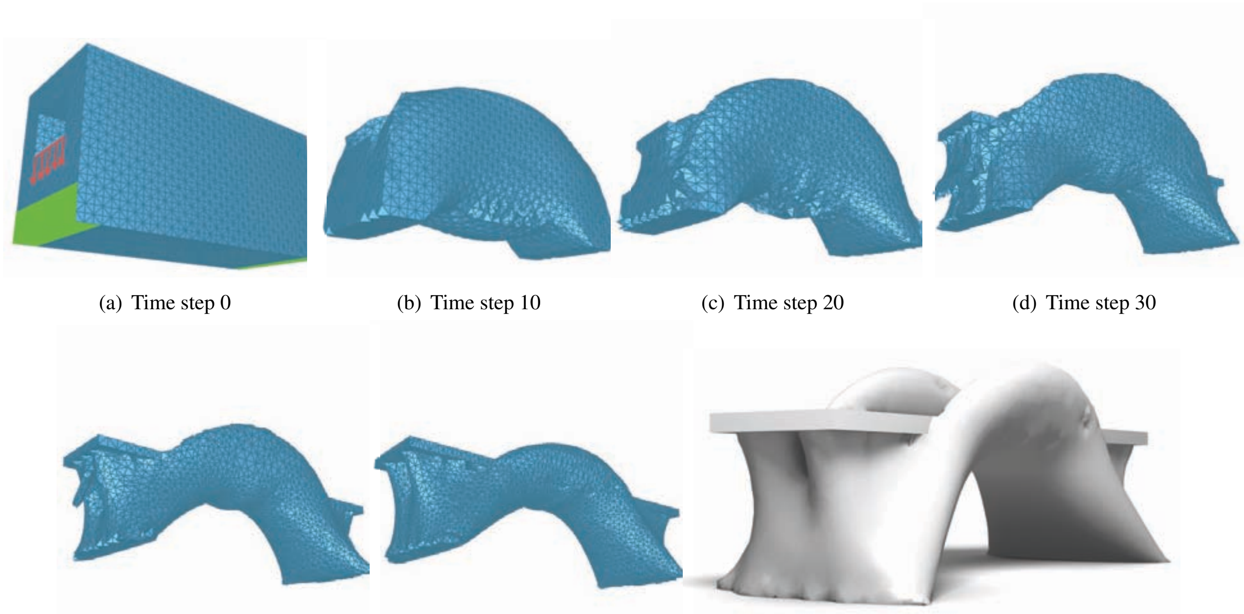
With the advances in 3D printing technology, topology optimization is not just of interest to manufacturers, but to anyone who has access to a 3D printer. Most consumers lack formal training in structural mechanics, which can hinder the process with many iterations and costly failed attempts. Consumers can under-engineer a design unsuitable for the intended load, or over-engineer a design that wastes expensive construction material. Topology optimization offers consumers a tool for designing shapes that meet their structural needs while using minimal construction resources.

In this paper, we present a fully automated design tool for designing structurally sound structures which can be manufactured, constructed or printed. The modeler only has to specify boundary conditions, the optimization objective, constraints and an initial structure. In other words, the designer specifies a set of requirements (the functionality of the structure and not the structure itself) and the method automatically designs a structure which fits those requirements. Note that this design process is significantly different from today where a designer manually models a structure and requirements are taken into account during this design process.

The proposed method for topology optimization is based on the Deformable Simplicial Complex (DSC) method [3]. The DSC method represents a solid structure with a conforming tetrahedral mesh (a simplicial complex) whose tetrahedral elements either lie entirely inside or outside the structure. The interface between solid and void (the surface) is represented explicitly by the triangular faces shared by an interior and exterior tetrahedral element. Furthermore, the DSC method ensures well-formed tetrahedral elements by constantly performing mesh improvement routines while the surface is being deformed. Finally, it provides adaptive resolution, allowing fine details where and when needed.

The method uses two optimization strategies:

Discrete optimization



(a) Time step 0 (b) Time step 10 (c) Time step 20 (d) Time step 30

(e) Time step 50 (f) Time step 70 (g) Final result

Figure 1: Given a few input parameters, the proposed method automatically optimizes the shape and topology of a 3D structure. Here is an example of optimizing a bridge. The initial structure is seen to the upper left along with supports (green) and loads (red). This structure is optimized such that stiffness is maximized and the amount of material is minimized. A few iterations of the method are depicted along with the result.

Relabels elements from solid to void to improve the objective or constraints which are not satisfied. The relabeling is based on topological derivatives [4][5][6][7][8], i.e. the change in the objective or constraints by introducing an infitesimal hole.

Continuous optimization

Performs a non-parametric shape optimization [9][10][11][12][13]. First, an improved shape, which is within a small perturbation of the current shape, is found by solving a constrained optimization problem using the Method of Moving Asymptotes (MMA) [14]. The surface is then deformed to this improved shape using the DSC method [3]. While the surface is deformed, the mesh is adapted such that its tetrahedral elements are well-formed at all times.

These optimization strategies are iterated until changes are small. An example is seen in Figure 1.

We will show that this tool is of interest to both engineers and designers. For example, we show that it can be used to improve stiffness and balance of a 3D model, to save material and to generate functional as well as, in our opinion, visually pleasing designs.

1.1. Related work

Recent trends in the computer graphics society are to add mechanical properties to 3D models. Prévost et al. have been concerned with the balance of printed models [15], Skouras et al. about printing deformable characters using a stiff and soft material [16] and several research teams have focused on self-supporting masonry structures [17][18][19].

A major concern has been to improve the stiffness of 3D models. Umetani et al. perform a cross-sectional structural analysis and visualize the result [20]. A user can then manually edit the model to improve the stiffness while getting almost instant feedback. The instant feedback is only possible because the analysis is limited to cross-sections. Stava et al. presents a more automated method for improving stiffness [21]. They perform a complete worst-case structural analysis on a tetrahedral mesh to determine the structurally weak regions. Based on this analysis, it is decided whether to improve the model by thickening, hollowing or adding a strut. Finally, Zhou et al. [22] also perform a worst-case structural analysis with more precise determination of the worst-case loads than in [21]. Furthermore, they conclude that solving a shape optimization problem to minimize stress is impractical due to the non-linearity and non-convexity of the problem. Therefore, they make do with visualizing the structurally weak regions.

Topology optimization problems are indeed non-convex. However, the topology optimization community has been solving these problems to at least local optimality for decades and the resulting designs usually perform better than designs optimized by humans [2]. Feasible solutions to these problems are often found by standard numerical gradient-based optimization algorithms. However, note that the smooth compliance functional is often chosen as the objective function to ease the optimization instead of the non-smooth, but often more interesting, maximal stress as Zhou et al. propose.

A key ingredient in a topology optimization method is the shape representation which is required to be able to handle topology changes. Hence, topological optimization has focused primarily on implicit representations over uniform voxel grids. Such representations can handle topology changes but lead to fixed-resolution results with cuberille artifacts. The most popular implicit topology optimization approaches are the density and level set approaches. The density approach [23][2] represents the structure by assigning a density value between 0 (void) and 1 (material) to each cell in a fixed grid or mesh. The structure is now deformed by changing these density values. The level set approach uses the level set method [24] evaluated on a fixed grid or mesh [25][26]. Here, the structure is represented by the zero level set and deformed by changes to the level set function. Both methods iteratively change the shape to approach the optimum.

We propose to represent the surface explicitly. An explicit representation, for example a triangle mesh, has previously been used for shape optimization [9][10]. However, shape optimization does not allow for topology changes and often only small shape deformations. Furthermore, it has been used in combination with the level set method [27][28][29][30][31] where it is necessary to constantly switch between the implicit and explicit representations. An explicit representation has also been used in combination with a computationally expensive remeshing of the entire design domain at each iteration [4][32]. Finally, it has previously been shown that using the DSC method for topology optimization works in 2D and therefore has potential [33]. However, here, we show that this concept is able to solve real-world topology optimization problems in 3D.

Note that this list of structural optimization methods is far from exhaustive.

1.2. Contributions

The main contributions of this paper are as follows.

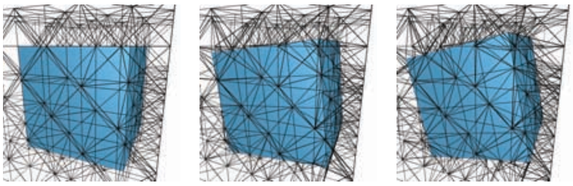
- As opposed to previous methods introduced in computer graphics, our method automatically optimizes the shape and topology of a structure given boundary conditions, an objective function, constraints and an initial shape. This completely eliminates the manual editing which has been characteristic for the current approaches.

- Compared to current methods from the topology optimization community, the method uses a single explicit representation to represent the structure and, at the same time, is able to handle topology changes. This gives rise to several advantages including a single mesh for shape representation and finite element calculations, possibility of both continuous and discrete optimization strategies and both the initial and optimized structure are in the form of surface triangle meshes. Finally, the adaptive mesh makes it possible to achieve a much more detailed result within reasonable time on an ordinary laptop than otherwise possible using the standard fixed grid methods.

- To be able to solve real-world topology optimization problems in 3D, it was necessary to make significant changes compared to the 2D proof-of-concept by Christiansen et al. [33]. Consequently, the discrete step relabels elements based on an optimization procedure which takes constraints into account instead of based on a simple threshold of the objective. Furthermore, the presented method handles self weight, it is initialized by any surface triangle mesh, areas can be fixed to either solid or void and several global constraints have been implemented and utilized. Finally, the requirements for computational efficiency is much higher in 3D than 2D. Therefore, the mesh adaptivity of the DSC method is utilized and the computations are distributed on multiple cores.

2. Method

The proposed method uses a simplicial complex to represent the shape of a structure. A simplicial complex discretizes a domain into tetrahedral elements. In 3D it consists of the simplices; nodes (points), edges (line pieces), faces (triangles) and tetrahedra (triangular pyramids). Furthermore, the tetrahedra do not overlap and any point in the discretized domain is either inside



(a) Time step 1 (b) Time step 2 (c) Time step 3

Figure 2: Rotation of a cube using the Deformable Simplicial Complex method. The interface between solid and void (the surface of the cube) is depicted in turquoise. Furthermore, all edges of the simplicial complex are drawn in black.

(a) Time step 1 (b) Time step 2 (c) Time step 3

Figure 3: Illustration of topology changes using the Deformable Simplicial Complex method. Here, only edges having both end nodes on the surface are drawn. As the objects approach each other the tetrahedra between the objects get squeezed. When a tetrahedron between the two surfaces is squeezed too much, this tetrahedron will be collapsed. Consequently, the only thing separating the two objects is a face. However, this face has tetrahedra which are labeled solid on both sides and it is therefore no longer part of the surface. Consequently, the two objects are now merged into one.

a tetrahedron or on the boundary between tetrahedra. In addition, all tetrahedra are labeled as being either void (no material) or solid (filled with material). Therefore, the interface between solid and void (the surface) is represented by the faces that are sandwiched between a tetrahedron labeled void and a tetrahedron labeled solid. Figure 2 depicts a cube represented by a simplicial complex. The tetrahedral mesh generator TetGen [34] is used to generate the initial mesh.

Apart from the shape representation, the tetrahedral elements of the simplicial complex can be used for physical computations using the finite element method. Since the finite element analysis will produce large errors if used with nearly degenerate tetrahedra, it is important to sustain a high quality mesh.

2.1. Deformable Simplicial Complex method

To ensure a high quality mesh, we use the Deformable Simplicial Complex (DSC) method [3][1]. The DSC method ensures high quality tetrahedral elements during deformation of a model embedded in a simplicial complex as illustrated in Figure 2. Low quality tetrahedra (slivers, wedges, caps and needles) are removed by continuously performing a set of mesh operations while the surface is being deformed. The tetrahedron quality measure is $\frac{6\sqrt{2}V}{(\frac{1}{6}\sum_i l_i^2)^{3/2}}$ [35] where V is the volume of the tetrahedron and l_i is the length of edge i. Note that the DSC method only improves the mesh quality where necessary (often near the surface). Furthermore, the DSC method also handles topology changes by removing low quality tetrahedra which are sandwiched between two surfaces. This is illustrated by two objects colliding in Figure 3.

[1]An open-source framework is available at `www.github.com/asny/DSC`

In addition to ensuring high quality tetrahedral elements, the DSC method also controls the level of detail of both the surface and the tetrahedral mesh. In practice, the DSC method attempts to collapse too small simplices and split too large simplices. Consequently, we always attain a mesh of the desired complexity, described by the discretization parameter δ (corresponding to the average edge length). More importantly, the detail control allows for mesh adaptivity. This means that smooth regions on the surface are represented by a more coarse discretization than regions with small features.

The mesh operations used are *smoothing* [36] (not performed on surface nodes), *edge split* [37], *edge collapse* [37], *edge removal* [38] and *multi-face removal* [38]. The latter two use the flips illustrated in Figure 4. Consequently, these two mesh operations do not change the position of any nodes, only the connectivity. The quality of the mesh is improved by all five operations, whereas the detail level of the mesh is controlled through the operations edge split and edge collapse. Note that changes have been made compared to [3]. The multi-face retriangulation, optimization-based smoothing, null-space smoothing and tetrahedron relabeling operations have not been necessary for this application. Removing these operations has resulted in a significant speed-up. Also, the edge removal operation on the surface and boundary is an addition since [3].

The strategy for moving the surface nodes is to first compute a destination p_n^* for each surface node n currently at position p_n. The destination p_n^* is computed using a user-defined velocity function which, for the case of topology optimization, will be described later. Afterwards, all surface nodes are moved from p_n to p_n^* using the strategy illustrated in Figure 5.

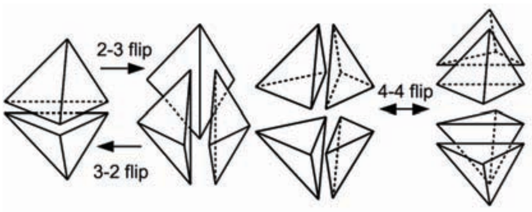


Figure 4: Illustrations of 2-3, 3-2 and 4-4 flips inspired by the illustration in [38].

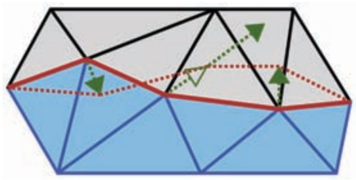


Figure 5: Illustration of how the surface (red) is moved in 2D. The same principle applies to 3D. A filled arrow indicates the destination $\boldsymbol{p}_n^*$ of the surface node n. One of the nodes cannot move to its destination without creating low quality tetrahedra and it is therefore only moved as depicted by the unfilled arrow. The other two are moved to their destinations. Then, mesh operations are applied to improve the mesh quality and the node that did not reach its destination is moved again. This is repeated until all nodes have reached their destinations.

2.2. *Structural analysis*

In this paper, we will optimize the topology of physically valid structures in static equilibrium. In order to achieve physical validity, structural analyses using the finite element method are performed. This implies considering the *discretization*, *boundary conditions* and *equilibrium* which are the topics of this section.

As described previously, a domain is discretized into high quality tetrahedral elements which are analyzed using the finite element method. Using quadratic basis functions solves a well-known issue with a jagged surface when using the analysis as a basis for non-parametric shape optimization [11][12]. Consequently, quadratic basis functions are chosen instead of linear to interpolate the tetrahedral elements. Therefore one control point c is associated with each node and edge of a tetrahedron. Furthermore, the positions of all control points are assembled in a vector termed $\boldsymbol{p} = [\ldots, \boldsymbol{p}_c^T, \ldots]^T$. In addition, each tetrahedron t has an associated material m_t with material parameters *density* ρ_t, *Young's modulus* E_t and *Poisson's ratio* ν_t. Finally, the materials of the tetrahedra are also assembled in a vector $\boldsymbol{m} = [\ldots, m_t, \ldots]^T$.

The local *stiffness matrix* $\boldsymbol{K}_t$ contains information on the stiffness of tetrahedron t. It depends on both the positions of the control points $\boldsymbol{p}$ and the materials of the tetrahedra $\boldsymbol{m}$ and can be calculated by

$$\boldsymbol{K}_t(\boldsymbol{m}, \boldsymbol{p}) = \int_{V_t} \boldsymbol{B}_t^T(\boldsymbol{p}) \boldsymbol{E}_t(\boldsymbol{m}) \boldsymbol{B}_t(\boldsymbol{p}) \, \partial(x, y, z) \tag{1}$$

We have chosen only to consider isotropic linear materials. Consequently, the constitutive matrix $\boldsymbol{E}_t(\boldsymbol{m})$ which relates stress and strain is

$$\boldsymbol{E} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$$

where $\boldsymbol{E}_t(\boldsymbol{m})$ is shortened to $\boldsymbol{E}$, $E_t(\boldsymbol{m})$ to E and $\nu_t(\boldsymbol{m})$ to ν. Finally, the strain-displacement matrix $\boldsymbol{B}_t(\boldsymbol{p})$ is related to the shape of the tetrahedron and the basis functions. For more details, see a text book on the finite element method used for structural analysis, e.g. [39]. The global stiffness matrix $\boldsymbol{K}(\boldsymbol{m}, \boldsymbol{p})$ can then be assembled from the local stiffness matrices $\boldsymbol{K}_t(\boldsymbol{m}, \boldsymbol{p})$. Note that for elements with void as the associated material, $\boldsymbol{K}_t$ is not defined. Consequently, the void elements are eliminated from the finite element analysis, which decreases computation time.

In this paper, we will limit ourselves to static problems subject to a single load case. These problems are modeled by supports and external forces $\boldsymbol{f}_c$ which are both applied to the surface of the structure. In addition to external forces, the weight of the structure will cause gravitational forces

$$\boldsymbol{w}_c(\boldsymbol{m}, \boldsymbol{p}) = \boldsymbol{g} \sum_{i \in c} a_i \quad \rho_i(\boldsymbol{m}) \quad V_i(\boldsymbol{p}) \tag{2}$$

Here, $\boldsymbol{g} = [0, -9.8, 0]^T m/s^2$ is a vector of the gravitational acceleration and a_i is a scale factor computed by a mass lumping scheme for each element i. Furthermore, ρ_i is the density and $V_i(\boldsymbol{p})$ is the volume of tetrahedral element i which is adjacent to control point c. Consequently, the global *force vector* is

$$\boldsymbol{f}(\boldsymbol{m}, \boldsymbol{p}) = [\ldots, \boldsymbol{f}_c^T + \boldsymbol{w}_c^T(\boldsymbol{m}, \boldsymbol{p}), \ldots]^T \tag{3}$$

Since we desire a structure in static equilibrium, the sum of the forces on all particles must be zero (Newton's first law). Consequently, we will utilize the equilibrium equations

$$\boldsymbol{K}(\boldsymbol{m}, \boldsymbol{p})\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{m}, \boldsymbol{p}) \tag{4}$$

These equations are used to calculate the global *displacement vector* $\boldsymbol{u} = [\ldots, \boldsymbol{u}_c, \ldots]$. At each control point c, $\boldsymbol{u}_c$ represents the displacement caused by the forces $\boldsymbol{f}$ applied to the structure. Note that, since $\boldsymbol{K}$ and $\boldsymbol{f}$ are functions of $\boldsymbol{p}$ and $\boldsymbol{m}$, so is $\boldsymbol{u}$.

Solving the equilibrium equations is the most time consuming part of the optimization. Furthermore, the number of equations scales linearly with the number of degrees of freedom. Consequently, the sparse solver *CHOLMOD* [40], which is a part of the *SuiteSparse* library [41], is used to solve the equilibrium equation efficiently using multiple cores.

2.3. Optimization

We want to optimize an objective function f by changing the shape and topology of the structure. Therefore, the objective can be anything as long as it is a function of the shape and topology. Furthermore, there are two ways to change the shape and topology. The first is to change the position $\boldsymbol{p}_n$ of a design node n, the other is to change the material m_e of a design element e. A node is a design node n if it is

- on the surface of the structure,
- not supported,
- not subjected to any external forces and
- not part of a fixed domain (see Section 2.5).

Furthermore, a tetrahedral element is a design element e if it is

- solid,
- not adjacent to a control point subjected to external forces and
- not part of a fixed domain (see Section 2.5).

For the test cases presented here, we seek to find the structure which is as stiff as possible. Consequently, the objective function is compliance

$$f(\boldsymbol{m},\boldsymbol{p}) = \boldsymbol{u}^T\boldsymbol{K}(\boldsymbol{m},\boldsymbol{p})\boldsymbol{u} \tag{5}$$

Note that since this objective is a function of the displacements $\boldsymbol{u}$, we need to solve Equation 4 to evaluate it. The reason for choosing to minimize compliance and not for example maximal Von Mises stress is that the compliance function is smooth. This is a significant advantage for the optimization algorithm. However, we plan to minimize the maximal Von Mises stress using the same method in the future.

It is often desirable to constrain the optimization. In some test examples, we choose to limit the amount of material used, i.e. the optimization is subject to a global volume constraint:

$$g_1(\boldsymbol{m},\boldsymbol{p}) = \frac{V(\boldsymbol{m},\boldsymbol{p})}{V^*} - 1 \tag{6}$$

Where $V(\boldsymbol{m},\boldsymbol{p})$ is the total volume of the solid elements and V^* is the maximum volume of the structure.

Optimized results are often not manufacturable. For example, the optimized results often contain many details. A partial remedy is to constrain the total surface area, called a perimeter constraint [42].

$$g_2(\boldsymbol{m},\boldsymbol{p}) = \frac{A(\boldsymbol{m},\boldsymbol{p})}{A^*} - 1 \tag{7}$$

Here, $A(\boldsymbol{p})$ is the total area of triangles sandwiched between a void and a (not fixed) solid element and A^* is the maximum surface area allowed. This constraint enforces a smoothness of the surface and thereby to some degree prevents small details and thin plates. However, since it is a global constraint, these undesirable features are not guaranteed to be eliminated.

Finally, in some cases, we want to limit the possible change from the original shape. In these cases, the original design nodes are added to a set O. If, during the optimization, an edge connecting two original nodes is split, the new node will be added to the set. However, if a hole appears inside the structure, the nodes on that internal surface are not added. Furthermore, the original surface is stored such that the distance $d_n(\boldsymbol{m},\boldsymbol{p})$ from $n \in O$ to the original surface can be calculated. Finally, the function $t_n(\boldsymbol{m},\boldsymbol{p})$ computes the distance from $n \notin O$ to the surface represented by the nodes in the set O. In other words, this function calculates the thickness of the shell of the structure. We can now limit the change from the original surface as well as ensuring that holes will not appear in this surface by applying the constraint:

$$\begin{aligned} g_3(\boldsymbol{m},\boldsymbol{p}) =& \frac{1}{N_{\in O}}\sum_{n\in O}\max(d_n(\boldsymbol{m},\boldsymbol{p}) - D^*, 0)^2 \\ &+ \frac{1}{N_{\notin O}}\sum_{n\notin O}\max(T^* - t_n(\boldsymbol{m},\boldsymbol{p}), 0)^2 \end{aligned} \tag{8}$$

Here, D^* is the maximal change from the original surface and T^* is the minimum thickness of the shell of the structure. Note that g_3 is C^1 continuous and thereby differentiable.

2.3.1. Continuous optimization

The first part of the optimization procedure is to locally perturb the surface of the structure such that it iteratively gets closer to optimum. This part of the optimization procedure consists of calculating an improved

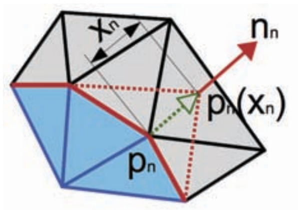


Figure 6: Illustrates the destination $\boldsymbol{p}_n(x_n)$ of node n as a function of the design variable x_n. Furthermore, $\boldsymbol{p}_n$ is the current position and $\boldsymbol{n}_n$ is the normal.

position $\boldsymbol{p}_n^*$ for each design node n. Afterwards, the structure is deformed by moving each design node from its current $\boldsymbol{p}_n$ to the more optimal position $\boldsymbol{p}_n^*$ as described in Section 2.1. Note that since the DSC method handles topology changes, these can occur. Thin structures can collapse and holes can disappear. However, holes will not appear inside the structure during this step. Also, note that the material parameter $\boldsymbol{m}$ is fixed during this step.

Moving the design nodes in the tangent directions will not change the surface much. Consequently, each design node n is associated with one design variable only. A design variable x_n represents the distance node n is moved in the normal direction $\boldsymbol{n}_n$ from the current position $\boldsymbol{p}_n$ as illustrated in Figure 6. The design variables are assembled in the vector $\boldsymbol{x} = [\ldots, x_n, \ldots]^T$. Consequently, the positions of the control points as a function of the design variables can be expressed as $\boldsymbol{p}(\boldsymbol{x})$.

The relation between the current position $\boldsymbol{p}_n$, the optimized position $\boldsymbol{p}_n^*$ and the optimized design variable x_n^* for a design node n is

$$\boldsymbol{p}_n^* = \boldsymbol{p}_n(x_n^*) = \boldsymbol{p}_n + x_n^* \ \boldsymbol{n}_n \tag{9}$$

To estimate $\boldsymbol{x}^* = [\ldots, x_n^*, \ldots]^T$, a smooth non-linear optimization problem is solved:

$$\begin{aligned} \boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} &: f(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x})) = \boldsymbol{u}^T \boldsymbol{K}(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x}))\boldsymbol{u} \\ subject\ to &: g_i(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x})) \leq 0, \ i = 1, 2, 3 \\ &: \boldsymbol{K}(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x}))\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x})) \\ &: \boldsymbol{x}^{\min} \leq \boldsymbol{x} \leq \boldsymbol{x}^{\max} \end{aligned} \tag{10}$$

Here, $\boldsymbol{x}^{\min} = [\ldots, x_n^{\min}, \ldots]^T$ and $\boldsymbol{x}^{\max} = [\ldots, x_n^{\max}, \ldots]^T$ are move limits on the design variables $\boldsymbol{x}$. Generally, $\boldsymbol{x}^{\min}$ and $\boldsymbol{x}^{\max}$ are chosen such that the design nodes will not create degenerate tetrahedra during the optimization. Consequently, the new shape can only be a small perturbation from the current shape and Equation 10 will be solved many times. Furthermore, the move limits ensure that the design nodes stay inside a user-specified design domain. Therefore, the structure cannot extend beyond the boundaries of this design domain.

We use the gradient-based optimization algorithm Method of Moving Asymptotes (MMA) [14] to solve the optimization problem in Equation 10. This is an iterative optimization procedure which is stopped when the infinity norm of the change in $\boldsymbol{x}$ is less than a threshold or at iteration 5. In addition to evaluating the objective function and constraints, the derivatives of these functions with respect to each of the design variables x_n have to be evaluated at each iteration. Computing $\frac{\partial}{\partial x_n}\boldsymbol{u}$ is not efficient. However, using the adjoint variable method (utilizing the equilibrium equations) [43][44], we get an analytical expression for $\frac{\partial}{\partial x_n} f(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x}))$ without the problematic term $\frac{\partial}{\partial x_n}\boldsymbol{u}$:

$$\begin{aligned} &\frac{\partial f(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x}))}{\partial x_n} = \\ &-\boldsymbol{u}^T \frac{\partial \boldsymbol{K}(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x}))}{\partial x_n}\boldsymbol{u} + 2\boldsymbol{u}^T \frac{\partial \boldsymbol{f}(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x}))}{\partial x_n} \end{aligned} \tag{11}$$

Still, since the equilibrium equations have to be evaluated at each iteration, this continuous optimization step is the most expensive part of the optimization procedure.

2.3.2. *Discrete optimization*

In addition to changing the shape by moving the design nodes, a discrete optimization step is performed where the materials $\boldsymbol{m}$ are changed and the positions $\boldsymbol{p}$ are not. The step has two purposes; introducing holes inside the structure and increasing the convergence rate of the continuous optimization. The optimization problem can be written as

$$\begin{aligned} \boldsymbol{m}^* = \arg\min_{\boldsymbol{m}} &: f(\boldsymbol{m}, \boldsymbol{p}) = \boldsymbol{u}^T \boldsymbol{K}(\boldsymbol{m}, \boldsymbol{p})\boldsymbol{u} \\ subject\ to &: g_i(\boldsymbol{m}, \boldsymbol{p}) \leq 0, \ i = 1, 2, 3 \\ &: \boldsymbol{K}(\boldsymbol{m}, \boldsymbol{p})\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{m}, \boldsymbol{p}) \\ &: m_e \in \{void, solid\} \end{aligned} \tag{12}$$

Note that the set of possible materials is limited to void and solid. However, it is possible to extend this approach to handle multiple materials. Furthermore, we choose that only solid elements are design elements. Consequently, this step only removes material from the structure. If it removes material near the surface, this will speed up shape changes. On the other hand, if it removes material inside the structure, a hole is created.

The discrete optimization problem in Equation 12 is NP-hard. However, since this optimization problem is combined with a continuous optimization, it is not necessary to solve it to optimality. Consequently, this step will seek to improve the objective while trying to satisfy the constraints by relabeling tetrahedra. The relabeling will be based on discrete derivatives, i.e. the change in objective or constraints when changing the material in element e from solid to void:

$$\Delta_e f(\boldsymbol{m}, \boldsymbol{p}) = f(\boldsymbol{m}_e^v, \boldsymbol{p}) - f(\boldsymbol{m}, \boldsymbol{p}) \tag{13}$$

$$\Delta_e g_i(\boldsymbol{m}, \boldsymbol{p}) = g_i(\boldsymbol{m}_e^v, \boldsymbol{p}) - g_i(\boldsymbol{m}, \boldsymbol{p}), \; i = 1, 2, 3 \tag{14}$$

Here, $\boldsymbol{m}_e^v$ equals $\boldsymbol{m}$ where m_e is void instead of solid. However, computing these discrete derivatives for compliance is inefficient since the equilibrium equations then have to be evaluated once for each solid tetrahedron. Instead, we will use an approximation based on the theory of topological derivatives [4][5][45][6]. The topological derivative corresponds to the influence on the objective function of introducing an infinitesimal hole in element e. For compliance, the discrete derivative can therefore be approximated by

$$\Delta_e f(\boldsymbol{m}, \boldsymbol{p}) \approx 3\boldsymbol{u}^T \boldsymbol{K}_e(\boldsymbol{m}, \boldsymbol{p})\boldsymbol{u} - \frac{2V_e(\boldsymbol{p})}{N_{\in e}} \sum_{c \in e} \boldsymbol{u}_c^T \boldsymbol{g} \tag{15}$$

The first part of the optimization strategy is to improve the objective function while decreasing or satisfying all constraints. A constraint i is decreased if

$$\Delta_e g_i(\boldsymbol{m}, \boldsymbol{p}) \leq 0 \tag{16}$$

and satisfied if

$$g_i(\boldsymbol{m}, \boldsymbol{p}) + \Delta_e g_i(\boldsymbol{m}, \boldsymbol{p}) \leq 0 \tag{17}$$

Hence, a design element e is relabeled from solid to void if either of equations 16 and 17 are satisfied for all constraints and

$$\Delta_e f(\boldsymbol{m}, \boldsymbol{p}) < 0 \tag{18}$$

The second part of the optimization is to try to improve constraints which are not satisfied. Therefore, if constraint i is not satisfied, i.e. $g_i(\boldsymbol{m}, \boldsymbol{p}) > 0$, we will try to find an optimal design element e^* to relabel from solid to void. Noting that $\Delta_e f(\boldsymbol{m}, \boldsymbol{p}) \geq 0$, the optimal design element e^* is found by solving

$$e^* = \arg\min_e - \frac{\Delta_e f(\boldsymbol{m}, \boldsymbol{p})}{\Delta_e g_i(\boldsymbol{m}, \boldsymbol{p})} \tag{19}$$

where all arguments e satisfy

$$\Delta_e g_i(\boldsymbol{m}, \boldsymbol{p}) < 0 \tag{20}$$

and either Equation 16 or 17 for all constraints. Design element e^* is then relabeled from solid to void. This process is repeated as long as constraint i is not satisfied and an optimal element e^* exists.

2.4. Disconnected material

The continuous and discrete optimization steps can very well result in material which is disconnected from the main structure. These parts do not contribute to the objective. Furthermore, since void elements are eliminated from the finite element analysis, disconnected material will result in the equilibrium equations not having a unique solution. Consequently, disconnected material is removed by performing a connected component analysis and making every component, except for the largest, void.

2.5. Initialization

To initialize the optimization, the user has to specify boundary conditions, an objective function, constraints and an initial structure.

The boundary conditions are the supports and external forces applied to the surface of the structure as described in Section 2.2. Furthermore, the boundaries of the design domain (the domain where material can reside) have to be specified. Finally, it is possible to specify fixed domains (areas that are either always solid or always void). The fixed void areas are implemented as not being a part of the design domain. However, the fixed solid domains are enforced by assigning a different label to the tetrahedra inside these domains. Consequently, an invisible surface exists between the fixed and non-fixed solid domains. The shape of this surface should not be changed in any way. However, we still want the DSC method to improve the mesh quality and control the level of detail at this surface. Consequently, the DSC method is modified such that only mesh operations which do not change the surface are performed at the surface between fixed and non-fixed domains.

In all of the example problems presented here, the objective is to minimize compliance since it is often desirable to produce as stiff a structure as possible. However, choosing another objective is as simple as changing the objective function and calculating the shape and topological derivatives of the new function. For example, the same approach has been used for balancing of 3D models [46]. Furthermore, different problems require different constraints. In this paper, we present several different global constraints to illustrate their effect on the design. The effect can be quite drastic and consequently the constraints are as important as the objective.

Finally, the initial model is a triangle mesh. Consequently, any surface mesh can be used as a starting point for the optimization without any conversions. In this paper, we choose to initialize the optimization by triangle meshes of existing models and by generated meshes that fill the entire design domain.

2.6. Method summary

The method consists of two steps:

Step 1: ***Discrete optimization***
Improves the objective as well as unsatisfied constraints by relabeling elements from solid to void based on their topological derivatives as described in Section 2.3.2. Then, removes disconnected material.

Step 2: ***Continuous optimization***
Solves the optimization problem in Equation 10 using the gradient-based optimization algorithm MMA (Section 2.3.1). MMA hereby estimates the optimal values of the design variables $\boldsymbol{x}^* = [\ldots, x_n^*, \ldots]^T$. Then, each design node n is moved from position $\boldsymbol{p}_n$ to $\boldsymbol{p}_n^* = \boldsymbol{p}_n + x_n^* \ \boldsymbol{n}_n$ using the DSC method as described in Section 2.1. Finally, disconnected material is removed.

These two steps make up one time step and are iterated until the changes on the surface from consecutive time steps are small.

Problems can arise if a volume or perimeter constraint is applied. The optimization will seek to obey the constraint before taking the objective into account. This can lead to undesired removal of material from places where it is necessary. Our solution to this problem is to gradually lower the constraint such that $V^*(t) = \max(\alpha^t, V^*)$ and $A^*(t) = \max(\beta^t, A^*)$ where t is the time step and $0 < \alpha < 1$ and $0 < \beta < 1$ are constants.

2.7. Efficiency

Efficiency is essential when performing topology optimization in 3D. A major piece of the puzzle to make this approach more efficient than standard fixed grid methods is to take advantage of the mesh adaptivity inherent to the DSC method. Consequently, the surface is represented by a fine discretization whereas large tetrahedra discretize parts far away from the surface. Furthermore, the main computational power should be used to achieve a fine resolution near the optimum. When the optimization is initialized by a 3D model, the optimum is assumed to be close. However, that is probably not the case when the optimization is initialized by filling the design domain with material. Consequently, in these cases, we slowly lower the discretization parameter δ by multiplying it by 0.99 at each time step. The detail control, described in Section 2.1, will then increase the mesh complexity. Note that this strategy is especially effective since the method only calculates on solid elements. However, solving the equilibrium equations is still the most time-consuming part. Consequently, we utilize multiple threads on the CPU to speed up these computations. Also, computing the gradients of the compliance function and assembling the global stiffness matrix K and force vector F are parallelized.

3. Results

The proposed method can be used in the fabrication design process in areas such as construction, manufacturing and design. In this section, we will illustrate this statement by solving problems within each of these fields. The results are generated on a laptop with a 2.4 GHz quad-core Intel Core i7 processor and 8 GB of 1333 MHz DDR3 RAM. Parameters and performance measures are depicted in Table 1. Furthermore, the objective of all examples is to minimize compliance subject to constraints as depicted in Table 1.

The raw surface triangle meshes of the optimized structures, i.e. the output as it looks from the optimization method, are visualised using Blender. No post processing like subdivision and smoothing has been utilized to improve the appearance. Furthermore, when material has been removed from inside a structure, the internal cavities are visualized by making the structure transparent. In addition to the optimized result, we will in some cases visualize the strain energy density (SED) at the surface of the final model. The SED depicts how much strain an element at the surface is subjected to. Here, the jet colormap is used, where blue and red depict low and high SED respectively. Furthermore, the SEDs are scaled between the minimum and maximum SED of the initial structure. Consequently, this visualizes how the stiffness has changed as a consequence of the optimization. In the same cases, we will also visualize the difference from the original model by a grayscale colormap. Here, gray means no change, darker means it has moved in the negative normal direction and lighter that it has moved in the normal direction. The distance is scaled by the largest change.

Problem	δ	V^* (α)	A^* (β)	D^*	T^*	f^*/f^0	Surface	Complex	Running time
	mm	% V^0 (-)	% A^0 (-)	% δ	% δ	-	# faces	# elements	minutes (#)
Bridge	423	20 (0.96)	30 (0.98)	-	-	304 %	9883	29836	68 (70)
Statue	50	50 (0.95)	-	15	100	27 %	35868	66314	275 (20)
Dinosaur	1.4	-	-	15	100	46 %	6876	15071	11 (5)
Armadillo	2.8	-	-	15	100	13 %	9872	15819	60 (50)
Table 1	42	15 (0.96)	30 (0.98)	-	-	2671 %	5492	11761	16 (100)
Table 2	62	15 (0.96)	35 (0.98)	-	-	964 %	3543	5521	13 (60)
Table 3	42	15 (0.96)	30 (0.98)	-	-	5929 %	5374	11759	20 (100)
Chair 1	21	12.5 (0.96)	25 (0.98)	-	-	1199 %	4413	7929	15 (100)
Chair 2	21	12.5 (0.96)	30 (0.98)	-	-	625 %	5527	9026	18 (100)
Chair 3	27	12.5 (0.96)	30 (0.98)	-	-	927 %	3382	4927	8 (75)
Support	655	20 (0.96)	20 (0.98)	-	-	17 %	15064	27120	109 (100)

Table 1: Method parameters and performance measures for all example problems. The displayed values are the values as they appear after the optimization. The V^* and A^* values are stated in percent of the initial volume V^0 and surface area A^0 respectively whereas D^* and T^* are stated in percent of the discretization parameter δ. Furthermore, f^0 and f^* are initial and final compliance respectively. Finally, the # in the right-most column is the number of time steps.

(a) Initial SED (b) Final SED (c) Change (d) Transparent statue (e) Statue

Figure 7: Topology optimized cow statue which show that the method can optimize stiffness while saving material.

3.1. Construction

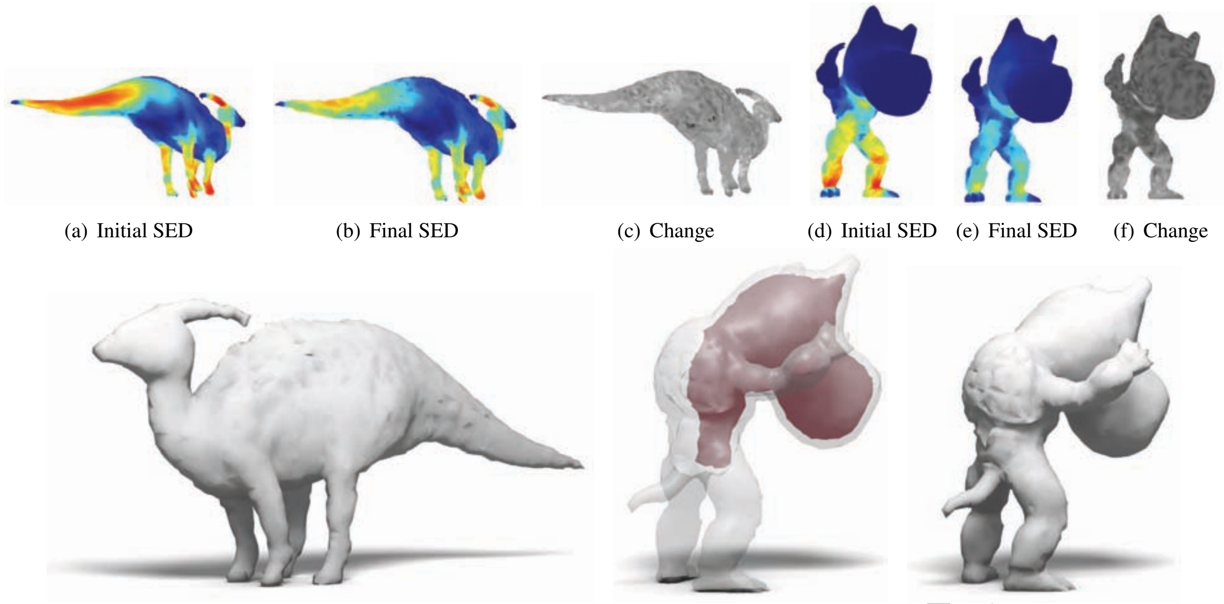
Topology optimization has traditionally been used for construction where the objective is to save material while ensuring stiffness. The presented method has the same capabilities as previous methods. Furthermore, it extends those methods by being able to initialize an optimization by a surface triangle mesh with no conversion necessary.

First, a bridge problem is initialized by a steel cube (30 15 12 m^3) with a space for vehicles and supports as depicted in Figure 1. The surface of the bridge is fixed and subjected to a distributed load pushing downwards (100 MPa). The result and optimization process are also depicted in Figure 1. The result shows that compliance has increased to 304% of the initial value during the optimization process. However, the optimized structure only uses 20% of the material used by the initial structure.

Next, a 4 m-long concrete statue is initialized by a 3D model of a cow (source: Aim@Shape). The statue is solid concrete, only subjected to gravitational forces and supported underneath all of its hoofs. The change in SED, shape changes and the optimized cow statue are depicted in Figure 7. This example shows that our method extends previous methods by being able to initialize an optimization by a 3D model (represented by a triangle mesh) without any conversion and, furthermore, remain close to this shape. Also, since the statue is subjected to gravitational forces only, compliance is improved at the same time as the amount of material is reduced.

3.2. Manufacturing

An important application of our method is as a tool to improve the stiffness of a given shape. Assume, we are given a 3D shape that is to be fabricated. The problem is to change the exterior shape as little as possible while using a minimum amount of material and ensuring that the fabricated object will be able to support itself and moreover withstand specified external loads. Further-



(a) Initial SED (b) Final SED (c) Change (d) Initial SED (e) Final SED (f) Change

(g) Dinosaur (h) Transparent Armadillo (i) Armadillo

Figure 8: Toy models optimized to improve both stiffness and balance while remaining close to the initial shape.

more, a side effect of optimizing a structure to bear its own weight is that the balance is improved.

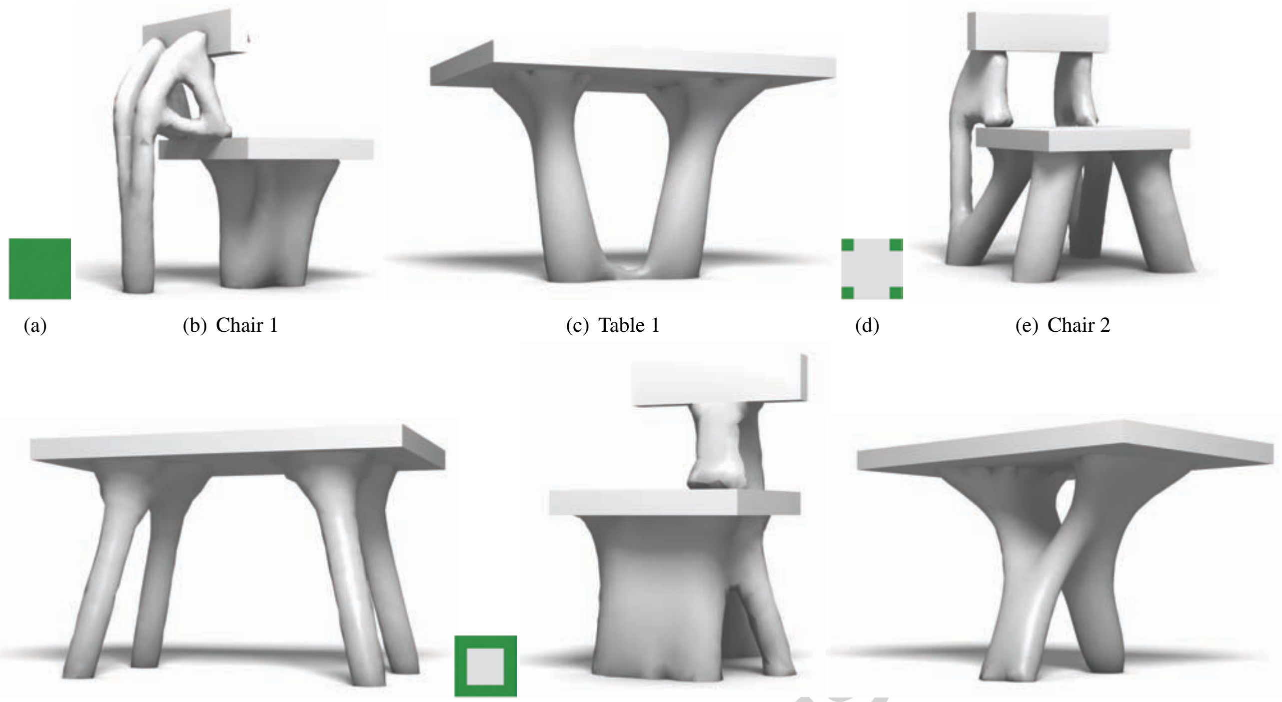
A 10 cm-long plastic model of a dinosaur (source: Aim@Shape) is subjected to external forces (5 MPa) on the tail and the head where one would expect the model to be weakest. Furthermore, each of the four feet are supported. The SEDs, shape changes and optimized dinosaur are depicted in Figure 8. Since the external forces are large compared to the gravitational forces, the optimization does not create any cavities. Instead, it redistributes material to places where it improves stiffness. Consequently, compliance is minimized to 46% of the initial value.

Next, a 10 cm-high plastic Armadillo model with a large head (source: Stanford University Computer Graphics Laboratory and edited in MeshMixer) is supported underneath both feet and only subject to gravity. The SEDs, shape changes and optimized model can be seen in Figure 8. It is evident that since the model has a large head it will lean forward and thereby subject the shins to large strain. When optimizing compliance, the strain is minimized and the balance of the model is improved as a side effect. However, since imbalance is not directly penalized by the objective function, balance is not guaranteed. A modification of the objective function or constraints would, however, guarantee balance by requiring the center of gravity to stay within the convex hull of the supports.

3.3. Design

When humans design a given 3D object, the main concerns are often to satisfy aesthetic and functional requirements. Topology optimization is not concerned with aesthetics but it satisfies functional requirements. However, topology-optimized shapes exhibit an organic and sparse feeling that is often visually pleasing. Therefore, such a tool is useful as part of a design workflow [47]. Furthermore, the method can be used to generate significantly different designs by slight changes to the input. This is significantly simpler for a designer than remodeling a surface.

Three plastic tables are modeled by a fixed layer of material at the top of a design domain (1.8 1.2 1.2 m^3) and a distributed load (2 MPa) pressing down on this layer. Furthermore, three chairs are initialized by filling a 0.6 0.8 0.6 m^3 design domain. The seat is modeled by a fixed void domain of size 0.4 0.4 0.4 m^3 and a fixed solid domain underneath which is subjected to a load (1 MPa). Finally, a backrest is modeled by a small fixed solid domain and subjected to a horizontal force (0.5 MPa). The difference between the problems are the position and extent of the supports. All supports are placed at the bottom of the design domain and have the shape depicted in figures 9(a), 9(d) and 9(g) as seen



(a) (b) Chair 1 (c) Table 1 (d) (e) Chair 2

(f) Table 2 (g) (h) Chair 3 (i) Table 3

Figure 9: Topology optimized tables and chairs which show the design capabilities of the suggested method. The difference between the problems are the supports (illustrated at the left of each row) and possibly the values of parameters. Note that the same illustration is used for both a table and a chair problem, therefore the dimensions of these illustrations are not correct.

from above. The optimized designs are depicted in Figure 9.

Finally, we will use the Qatar National Convention Center as an example of a real-world architectural design problem. The Convention Center has an impressive façade which is a roof supported by a concrete topology-optimized structure [47]. To model this, we take advantage of the symmetry and thereby only optimize a quarter of the structure (the symmetry axes are depicted in Figure 10(d)). Consequently, the problem is initialized by a 125 20 15 m^3 cube where the top layer (1 m) is fixed and solid. The structure is supported at the bottom in a half circular area (Figure 10(d)) and only subjected to gravity. The result can be seen in Figure 10(e) and, in addition, we illustrate in Figure 10 the effect of changing the parameter for the perimeter constraint. Note that the result is not expected to look like the Convention Center since [47] use different boundary conditions and do not specify material, objective and constraints.

4. Conclusion

The presented method is the first to optimize both the 3D shape and topology of a surface triangle mesh without the use of an implicit representation. This is achieved by embedding the triangle mesh in a simplicial complex and using the Deformable Simplicial Complex method. Consequently, the method accepts a surface triangle mesh as input and outputs another surface triangle mesh which is only different from the input mesh where it has been optimized. Furthermore, as opposed to standard fixed grid methods, our method makes it possible to generate detailed designs within reasonable time on an ordinary laptop.

We have shown that the method automatically generates designs which satisfy some user-defined structural requirements. However, note that the search space is limited by global constraints and that there is no guarantee that the global optimum is reached. The bridge and the cow statue show that material can be saved where it is expensive or inconvenient while maintaining or improving stiffness. The dinosaur and Armadillo models show that 3D models automatically can be made stiffer

(a) $A^* = 13\%$ (b) $A^* = 15\%$ (c) $A^* = 17\%$

(d) (e) Roof support ($A^* = 20\%$)

Figure 10: Topology optimized roof support, optimized using different values for the perimeter constraint. This problem is inspired by the real world problem of supporting the roof of the Qatar National Convention Center. The supports are placed as depicted in Figure 10(d) where also symmetry axes are visualized as black lines.

and more balanced, while retaining the shape. Finally, the tables, chairs and roof support show that functional and, in our opinion, visually pleasing designs can be achieved with little effort from a designer. This is far from an exhaustive list of problems that can be solved using the presented method. As mentioned, topology optimization has been used to solve a wide variety of problems. To solve these or other problems, one only needs to model the boundary conditions and choose the objective, constraints and an initial structure. However, more advanced problems might require additional work. For example implementing additional objective functions and constraints, handling multiple load cases, using an anisotropic material model, handling dynamic problems and taking non-linearity into account.

We have shown that furniture and support structures for buildings can be modeled by specifying a few input parameters. Furthermore, both the input and output models are in the form of a surface triangle mesh. Consequently, this tool has potential to be used for modeling for films, videogames and other offline productions in addition to designing physical structures, especially if performance and user friendliness are improved. To increase performance, one idea is to take full advantage of the parallel nature of the finite element computations by, for example, feeding the computations to the GPU. Furthermore, parallelization of the DSC method would be beneficial. Another idea is to take even further advantage of the mesh adaptivity by lowering the discretization parameter more wisely. To increase the user friendliness, automatic determination of worst-case loads could be useful to limit the amount of user input. Also, finding an alternative to the perimeter constraint would be desirable since it can limit the optimization and its parameter is unintuitive and difficult to choose. Finally, most designers want to influence the design regularly during the design process. Therefore, a workflow which includes user feedback and post processing is needed.

Acknowledgements

The authors appreciate the support from the Villum Foundation through the grant: "NextTop"

[1] M. P. Bendsøe, N. Kikuchi, Generating optimal topologies in structural design using a homogenization method, Computer Methods in Applied Mechanics and Engineering 71 (2) (1988) 197 – 224.

[2] M. P. Bendsøe, O. Sigmund, Topology Optimization - Theory, Methods, and Applications, second edition Edition, Springer Verlag, Berlin, 2003.

[3] M. K. Misztal, J. A. Bærentzen, Topology adaptive interface tracking using the deformable simplicial complex, ACM Transactions on Graphics 31 (3) (2012) No. 24.

[4] H. A. Eschenauer, V. V. Kobelev, A. Schumacher, Bubble method for topology and shape optimization of structures, Structural and Multidisciplinary Optimization 8 (1994) 42–51.

[5] J. Sokolowski, A. Zochowski, On the topological derivative in shape optimization, SIAM Journal on Control and Optimization 37 (4) (1999) 1251–1272.

[6] R. A. Feijóo, A. A. Novotny, E. Taroco, C. Padra, The topological derivative for the poisson's problem, Mathematical Models and Methods in Applied Sciences 13 (12) (2003) 1825–1844.

[7] S. Garreau, P. Guillaume, M. Masmoudi, The topological asymptotic for pde systems: The elasticity case, SIAM J. Control Optim. 39 (6) (2000) 1756–1778.

[8] F. de Gournay, G. Allaire, F. Jouve, Shape and topology optimization of the robust compliance via the level set method, ESAIM: Control, Optimisation and Calculus of Variations 14 (2008) 43–70.

[9] C. Le, T. Bruns, D. Tortorelli, A gradient-based, parameter-free approach to shape optimization, Computer Methods in Applied Mechanics and Engineering 200 (9-12) (2011) 985–996.

[10] S. Arnout, M. Firl, K.-U. Bletzinger, Parameter free shape and thickness optimisation considering stress response, Structural and Multidisciplinary Optimization 45 (6) (2012) 801–814.

[11] Y. Ding, Shape optimization of structures: a literature survey, Computers & Structures 24 (6) (1986) 985–1004.

[12] B. Mohammadi, F. O. Pironneau, Applied Shape Optimization for Fluids, Oxford University Press, 2009.

[13] D. Bucur, G. Buttazzo, Variational Methods in Shape Optimization Problems, Progress in Nonlinear Differential Equations and Their Applications, Birkhäuser, 2006.

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

[15] R. Prévost, E. Whiting, S. Lefebvre, O. Sorkine-Hornung, Make It Stand: Balancing shapes for 3D fabrication, ACM Transactions on Graphics (proceedings of ACM SIGGRAPH) 32 (4) (2013) 81:1–81:10.

[16] M. Skouras, B. Thomaszewski, S. Coros, B. Bickel, M. Gross, Computational design of actuated deformable characters, ACM Trans. Graph. 32 (4) (2013) 82:1–82:10.

[17] F. De Goes, P. Alliez, H. Owhadi, M. Desbrun, On the Equilibrium of Simplicial Masonry Structures, ACM Transactions on Graphics 32 (4).

[18] Y. Liu, H. Pan, J. Snyder, W. Wang, B. Guo, Computing self-supporting surfaces by regular triangulation, ACM Trans. Graph. 32 (4) (2013) 92:1–92:10.

[19] D. Panozzo, P. Block, O. Sorkine-Hornung, Designing unreinforced masonry models, ACM Transactions on Graphics (proceedings of ACM SIGGRAPH) 32 (4) (2013) 91:1–91:12.

[20] N. Umetani, R. Schmidt, Cross-sectional structural analysis for 3d printing optimization, in: SIGGRAPH Asia 2013 Technical Briefs, SA '13, ACM, New York, NY, USA, 2013, pp. 5:1–5:04.

[21] O. Stava, J. Vanek, B. Benes, N. Carr, R. Mech, Stress relief: Improving structural strength of 3d printable objects, ACM Trans. Graph. 31 (4) (2012) 48:1–48:11.

[22] Q. Zhou, J. Panetta, D. Zorin, Worst-case structural analysis, ACM Trans. Graph. 32 (4) (2013) 137:1–137:12.

[23] M. P. Bendsøe, Optimal shape design as a material distribution problem, Structural Optimization 1 (4) (1989) 193–202.

[24] S. J. Osher, R. P. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, 1st Edition, Springer, 2002.

[25] M. Wang, X. Wang, D. Guo, A level set method for structural topology optimization, Computer Methods in Applied Mechanics and Engineering 192 (1) (2003) 227–246.

[26] G. Allaire, F. Jouve, A.-M. Toader, Structural optimization using sensitivity analysis and a level-set method, Journal of Computational Physics 194 (1) (2004) 363–393.

[27] S.-H. Ha, S. Cho, Level set based topological shape optimization of geometrically nonlinear structures using unstructured mesh, Computers & Structures 86 (13-14) (2008) 1447–1455.

[28] G. Allaire, C. Dapogny, P. Frey, Topology and geometry optimization of elastic structures by exact deformation of simplicial mesh, Comptes Rendus Mathematique 349 (17-18) (2011) 999–1003.

[29] S. Yamasaki, T. Nomura, A. Kawamoto, K. Sato, S. Nishiwaki, A level set-based topology optimization method targeting metallic waveguide design problems, International Journal for Numerical Methods in Engineering 87 (9) (2011) 844–868.

[30] Q. Xia, T. Shi, S. Liu, M. Y. Wang, A level set solution to the stress-based structural shape and topology optimization, Computers & Structures 90 - 91 (0) (2012) 55–64.

[31] G. Allaire, C. Dapogny, P. Frey, A mesh evolution algorithm based on the level set method for geometry and topology optimization, Structural and Multidisciplinary Optimization 48 (4) (2013) 711–715.

[32] K. Maute, E. Ramm, Adaptive topology optimization, Structural optimization 10 (1995) 100–112.

[33] A. N. Christiansen, M. Nobel-Jørgensen, N. Aage, O. Sigmund, J. A. Bærentzen, Topology optimization using an explicit interface representation, Structural and Multidisciplinary Optimization 49 (3) (2014) 387–399.

[34] H. Si, TetGen: A quality tetrahedral mesh generator and a 3d delaunay triangulator (2013).
URL `http://wias-berlin.de/software/tetgen/`

[35] V. N. Parthasarathy, C. M. Graichen, A. F. Hathaway, A comparison of tetrahedron quality measures, Finite Elem. Anal. Des. 15 (3) (1994) 255–261.

[36] D. A. Field, Laplacian smoothing and delaunay triangulations, Communications in Applied Numerical Methods 4 (6) (1988) 709–712.

[37] J. A. Bærentzen, J. Gravesen, F. Anton, H. Aanæs, Guide to Computational Geometry Processing: Foundations, Algorithms, and Methods, Springer, 2012.

[38] J. R. Shewchuk, Two discrete optimization algorithms for the topological improvement of tetrahedral meshes, in: Unpublished manuscript, 2002.

[39] R. D. Cook, D. S. Malkus, M. E. Plesha, R. J. Witt, Concepts and Applications of Finite Element Analysis, John Wiley & Sons, 2007.

[40] Y. Chen, T. A. Davis, W. W. Hager, S. Rajamanickam, Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate, ACM Transactions on Mathematical Software 35 (3) (2008) 22:1–22:14.

[41] T. A. Davis, W. W. Hager, I. S. Duff, SuiteSparse (2013).
URL `http://www.cise.ufl.edu/research/sparse/SuiteSparse/`

[42] R. Haber, C. Jog, M. Bendsøe, A new approach to variable-topology shape design using a constraint on perimeter, Structural optimization 11 (1-2) (1996) 1–12.

[43] O. Pironneau, Optimal shape design for elliptic systems, in: R. Drenick, F. Kozin (Eds.), System Modeling and Optimization, Vol. 38 of Lecture Notes in Control and Information Sciences, Springer Berlin Heidelberg, 1982, pp. 42–66.

[44] P. W. Christensen, A. Klarbring, An Introduction to Structural Optimization, Solid mechanics and its applications, Springer, 2008.

[45] J. Céa, S. Garreau, P. Guillaume, M. Masmoudi, The shape and topological optimizations connection, Computer Methods in Applied Mechanics and Engineering 188 (4) (2000) 713–726.

[46] A. N. Christiansen, R. Schmidt, J. A. Bærentzen, Automatic balancing of 3d models, Computer-Aided Design (2014) to appear.

[47] M. Sasaki, T. Itō, A. Isozaki, Morphogenesis of flux structure, AA Publications, 2007.