L_0-Regularization based Material Design for Hexahedral Mesh Models

Haoxiang Li[1], Jianmin Zheng[2]

[1]Nanyang Technological University, Singapore, haoxiang002@e.ntu.edu.sg
[2]Nanyang Technological University, Singapore, asjmzheng@ntu.edu.sg

Abstract. The deformation behavior of a deformable part depends on its underlying material. Properly distributing heterogeneous elastic materials over an object is important in part design and becomes an active research topic in computer aided design and graphics. This paper considers the problem of how to design heterogeneous elastic materials over a hexahedral mesh model that commonly appears in computer-aided design and engineering applications. Existing approaches to solving the problem typically apply L_2 regularization that is good for smoothly distributed material. Considering that many real-world objects likely have sparse material distribution, we propose an optimization formulation with a carefully designed objective function and L_0 regularization. An iterative algorithm is presented to solve the L_0-optimization problem. The L_0 regularization encourages sparsity of the output material distribution, which may facilitate some approaches for digital material design in multi-material additive manufacturing. The experimental results show that the proposed method can output material distribution to produce the desired deformation behavior.

Keywords: Hexahedral mesh, Deformation, Material distribution, FEM, L_0-regularization, Nonlinear optimization
DOI: https://doi.org/10.14733/cadaps.2022.1171-1183

1 INTRODUCTION

The modeling of deformation of an object is a fundamental task in many applications. For example, in computer graphics the deformation is used to describe the modeling process or the animation procedure. In product design, the deformation is related to the function behavior of the target product. Note that the deformation behavior of an elastic object depends on its underlying material. Thus theoretically the deformation can be defined or controlled by specifying proper material distributions. However, directly setting materials over a 3D object for the desired deformation behavior is a tedious and challenging task.

In this paper, we propose to design the material distribution of an elastic object by specifying forces applied to some positions of the object and the respective displacement. Then the design problem is converted into an inverse problem of optimizing the material assignment for individual elements of the object to match the

specified deformation behavior. This idea is actually borrowed from Xu et al's work [28], which proposed an interactive approach using the force-displacement pair to design material of tetrahderal meshes. We extend their idea to hexahedral mesh models. Hexahedral mesh models are quite common in computer-aided design and engineering applications.

We also consider regularizing the material distribution with different patterns. Xu et al.'s work adopts L_2 regularization in material design, which is suitable for smoothly distributed material. If an object's material is sparsely distributed, which is common in real world applications, a quantization is needed as a post-process. We propose to use L_0 regularization in our design problem, aiming to output sparse material distribution, which is suitable for objects with sparsely distributed materials and may also benefit 3D printing. In fact, as pointed out in [16], recent multi-material additive manufacturing enables the fabrication of an object with deposition of different types of materials. However, the number of materials that can be deposited during the fabrication process is limited. Thus to generate digital material compositions that can be printed and be able to achieve the desired behavior, an exemplar-based approach is proposed in [16]. Our work uses L_0 regularization to encourage sparsity of the output material distribution, which can facilitate such exemplar-based approaches in digital material design for additive manufacturing.

In summary, the contributions of the paper are in two aspects:

- We propose a method to compute the distribution of Young's modulus for hexahedral mesh models, which can achieve desired deformation behavior.
- We formulate the problem as a minimization problem with L_0 regularization to encourage sparsity in material distribution and a numerical method is developed.

The rest of the paper is organized as follows. Section 2 reviews some related work. Section 3 presents our material design method for hexahedral mesh models, which includes the formulation of the problem and the algorithmic details. Section 4 reports the experimental results to demonstrate the effects of the proposed method. Section 5 concludes the paper.

2 RELATED WORK

This section briefly reviews physically-based deformation, material design of deformable objects, and regularized parameter distribution, which are relevant to our work.

Physically-based deformation: Physically-based deformation is a common deformation technique in computer graphics [19, 22]. It defines or simulates the deformation based on some equations or laws from physics and mechanics [24, 3, 13]. To compute the deformation in practice, numerical methods such as the finite element method (FEM) are often used, which discretize objects into basic elements. To simulate the mechanical properties of objects with inhomogeneous materials, various computational models and algorithms have been proposed [27, 26], examples of which are homogenization coarsening and material refining methods. The homogenization approaches [14, 9] use macroscopic materials to describe microscopic details. The material refining methods separate different materials so that each computational unit contains one material type, making it easier to describe material properties in the simulation process. Recently, data-driven approaches become popular to handle complicated nonlinear models [5].

Material design of deformable objects: The deformation behavior of an object depends on its underlying material properties. To create visually pleasing simulations, people studied the optimization of material distribution in the past few years. Starting with visual effect in real life, Hasan et al. [11] provided a goal-driven approach to design objects with ideal surface properties. Chen et al. [7] later proposed a numerical method to design the rest shape instead of material parameters. Morovič et al. [18] explored the possibility of designing color and mechanical variety with preference. Our work, however, focuses more on designing deformation behavior through optimizing material properties in each hexahedral element. In 2010, Bickel et al.[4] proposed a framework to design printed objects with expected mechanical behavior. After that, Skouras et al. [23]

presented a method to design actuated deformable characters, allowing users to animate the resulting figure. These methods assumed available materials in real life and used discrete optimization or interpolation applied to macro-level structures. In 2015, Xu et al. [28] proposed a material design method with material subspace acceleration, which allows the interactive design of material distribution for elastic objects using applied forces and displacements. Specifically, given applied forces and the desired displacements as input, the method optimizes the material distribution such that the displacements meet the desired ones. For this purpose, an optimization procedure with force-based objective function is designed [2]. Moreover, a strong regularization is introduced to improve the numerical stability and impose some constraints on the optimization variable, which may however result in less data fidelity.

Regularization of parameter distribution: In addition to making the deformable object meet the specified deformation, the distribution of the material parameters is preferably regularized. Bickel et al. [5] and Xu et al. [28] proposed smooth distribution of the material parameters, which was implemented using L_2 regularization in optimization. The L_2-optimization is easy to implement, but it is not very suitable for applications with the requirement of sparsely distributed material, such as $3D$ printing where only a limited number of materials are available. On the other hand, the sparsity of the material distribution can be implemented using the L_0 regularization. In fact, L_0 regularization has been widely used in some fields like image processing [29, 30, 20] and mesh processing [12]. However, to the best of our knowledge, no previous work has been proposed to use L_0 regularization in elastic material design and optimization.

3 PROPOSED METHOD

Our material design problem can be described as follows. Given a hexahedral mesh, the user defines the deformation behavior of the mesh by specifying applied forces at some vertices of the mesh and the corresponding displacements, and we want to find the distribution of Young's modulus over the mesh such that the deformation of the mesh caused by the prescribed forces matches the user specified displacements. Here we also assume a constant Poisson's ratio for all hexahedral elements. Thus the problem can be formulated as an optimization problem that finds the optimal Young's modulus. This section describes a finite element method based formulation and its numerical solution.

3.1 Construction of Finite Element Equations

Suppose the input hexahedral mesh has n vertices and m hexahedrons. We consider the first-order, linear isotropic hexahedral elements. For a hexahedron e, the element stiffness matrix K_e that relates forces and displacements is [15]:

$$K_e = \iiint_{\Omega^{(e)}} \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\mathrm{d}\Omega^{(e)} \tag{1}$$

where $\Omega^{(e)}$ is the element domain defined in the world coordinate system, D is the elasticity matrix to relate stress and strain, and B is the strain-displacement matrix. For isotropic linear material, matrix D can be expressed as

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

where E is Young's modulus and ν is Poisson's ratio. Matrix B is defined as

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 & \mathbf{B}_4 & \mathbf{B}_5 & \mathbf{B}_6 & \mathbf{B}_7 & \mathbf{B}_8 \end{bmatrix} \tag{3}$$

where

$$\mathbf{B}_i = \begin{bmatrix} \partial N_i/\partial x & 0 & 0 \\ 0 & \partial N_i/\partial y & 0 \\ 0 & 0 & \partial N_i/\partial z \\ 0 & \partial N_i/\partial z & \partial N_i/\partial y \\ \partial N_i/\partial z & 0 & \partial N_i/\partial x \\ \partial N_i/\partial y & \partial N_i/\partial x & 0 \end{bmatrix} \tag{4}$$

with eight shape functions N_i corresponding to the eight nodes or vertices of the hexahdron. Shape functions are used to determine state variables' value at any point of the element from the state variables' values at the nodes of the element.

Note that different hexahedral elements may have different shapes. To simplify the computation, we map each hexahedral element to a reference element that is a cube of size 2. Then Eq.(1) can be re-written as

$$K_e = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}|\mathbf{J}|\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta \approx \sum_{i=1}^{NG}\sum_{j=1}^{NG}\sum_{k=1}^{NG} \omega_i\omega_j\omega_k \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}|\mathbf{J}| \tag{5}$$

where ξ, η, and ζ represent the local coordinates with respect to the reference element, and J is the Jacobian matrix used to map the derivatives of shape functions from world coordinates to reference coordinates. In Eq.(5), the right part is an integration process for numerically computing the element stiffness matrix K_e. For simplicity, we use Gaussian integration where NG is the number of integration points.

By assembling individual K_e, we get the global stiffness matrix K. Given applied forces f, we can compute mesh displacements u by solving the following linear system:

$$K(E)u = f \tag{6}$$

which gives

$$u = K^{-1}(E)f \tag{7}$$

if E is known.

3.2 Construction of the Objective Function

Denote by $\bar{u}$ the user-specified displacements. Our goal is to design the Young's modulus distribution $E \in R^m$ such that the displacements computed from Eq.(5) match $\bar{u}$ as much as possible. Hence we construct the following minimization problem:

$$\min_{E} \frac{1}{2}\|Su - \bar{u}\|^2 + \mu R(E) \quad \text{s.t. } E \in [E_{\min}, E_{\max}] \tag{8}$$

where $u \in R^{3n}$ is the solution of Eq.(5), $S \in R^{3c\times 3n}$ is a selection matrix that maps the displacements of all vertices to the displacements of the user-selected vertices, c is the number of the selected vertices, $E_{\min} \in R^m, E_{\max} \in R^m$ are the lower and upper bounds of Young's modulus, $R(E)$ is a regularization term, and μ is the weight balancing the first term (i.e., fidelity) and the regularization term.

In the above formulation, we adopt the displacement-based objective function which is more accurate compared with the force-based objective function, as pointed out by Bickel et al [5]. While $K(E)$ is linear in E, the displacement vector u is not linear in E and thus the first term of Eq.(8) is not quadratic in E. In addition, we often specify a few vertices that are to be fixed during the deformation, which helps to remove unwanted degrees of freedom.

The regularization term is introduced to enforce the prior knowledge on the distribution of Young's modulus and improve numerical stability. Particularly, we formulate the regularization using the L_0 norm. The L_0 regularization encourages clustering of similar material in adjacent elements. Let N be the set of two indices of adjacent hexahedra. Then

$$R_{L_0} = \sum_{(i,j)\in N} \|E_i - E_j\|_0 \,. \tag{9}$$

In practice, a common constraint on Young's modulus is its validity (i.e., $E \geq 0$). Although a few elements or regions with negative Young's modulus might still be statically stable for the shape, the negative values often result in odd dynamics or may lead to a non-invertible stiffness matrix K. We propose to set box constraints of the optimization variables using $E_{\min}$ and $E_{\max}$. We think that enforcing box constraints is more favorable than just a non-negative constraint since it is rare that Young's modulus ranges from a very small to a very large value for a single object.

Note that our optimization is formulated in the full space, which facilitates imposing the box constraints. While subspace techniques $E = \Phi z$ ($\Phi \in R^{m\times r}$) are often used to speed up the calculation, the box constraints become complex or inefficient in the subspace. Particularly, when the number m of hexahedra is significantly larger than the number r of modes, the number of constraints will be much larger than the number of the optimization variables in the subspace, which causes the optimization in the subspace to be overly-constrained. Instead of directly enforcing such a constraint, Xu et al. [28] suggested adopting a regularization term with a large weight to alleviate the issue of negative Young's modulus. However, increasing the contribution of the regularization may lead to less accurate data fidelity, and moreover there is no guarantee to get rid of it by this approach.

3.3 L_0-Optimization

Note that the L_0 regularization term is non-differentiable. We cannot directly feed it into gradient-based optimizer. To efficiently solve the problem, we introduce the auxiliary variable $P_{i,j} = E_i - E_j$ and re-write Eq.(8):

$$\min_E \frac{1}{2}\|Su - \bar{u}\|^2 + \mu \sum_{(i,j)\epsilon N} \|P_{i,j}\|_0 \;\text{ s.t. }\; P_{i,j} = E_i - E_j \tag{10}$$

Since $P_{i,j} = E_i - E_j$ are equality constraints, we re-formulate the objective function via the augmented Lagrangian method:

$$\frac{1}{2}\|Su - \bar{u}\|^2 + \sum_{(i,j)\in N} \left(\mu \|P_{i,j}\|_0 + \frac{r_p}{2}\|E_i - E_j - P_{i,j}\|^2 + \lambda_{ij}\left(E_i - E_j - P_{i,j}\right)\right) \tag{11}$$

where r_p is the coefficient of the penalty term and $\lambda_{i,j}$ is the Lagrange multiplier. Then we use the alternating direction method of multipliers (ADMM) [6] to solve the problem. While other numerical methods can also be used to solve the problem, the ADMM has some advantage that the problem can be converted into iteratively solving two sub-problems, each of which is relatively easily solved. Specifically, we iteratively perform the following three steps:

1. First, we fix E and optimize the auxiliary variable P which is the collection of all $P_{i,j}$:

$$\arg\min_{P} \sum_{(i,j)\in N} \left(\mu||P_{i,j}||_0 + \frac{r_p}{2}\left\|E_i - E_j - P_{i,j}\right\|^2 + \lambda_{ij}\left(E_i - E_j - P_{i,j}\right)\right) \tag{12}$$

It's a hard threshold problem. The auxiliary variable P can be solved point-wisely. For each $P_{i,j}$, we can set $P_{i,j}$ equal to zero or not and compare corresponding function value. Then, we have the update strategy as: $P_{i,j} = E_i - E_j + \frac{\lambda_{ij}}{r_p}$ if $(E_i - E_j + \frac{\lambda_{ij}}{r_p})^2 > \frac{2\mu}{r_p}$ else $P_{i,j} = 0$.

2. Second, we fix the auxiliary variable P and optimize E. The L_0 term is no longer active in this step. Thus we solve

$$\arg\min_{E} \frac{1}{2}\|Su - \bar{u}\|^2 + \sum_{(i,j)\in N} \left(\frac{r_p}{2}\left\|E_i - E_j - P_{i,j}\right\|^2 + \lambda_{ij}\left(E_i - E_j - P_{i,j}\right)\right) \tag{13}$$

The current objective function is differentiable. We can find the solution by either nonlinear conjugate gradient or LBFGS method with box constraints.

3. In the third step, we update Lagrange multiplier λ_{ij} in the same way as the original ADMM algorithm. For each λ_{ij}, we have:

$$\lambda_{ij} \leftarrow \lambda_{ij}^k + r_p(E_i - E_j - P_{i,j}) \tag{14}$$

It is worth pointing out that here we design the update sequence in a way such that the effect of the deviation phenomenon is reduced. Otherwise, if we first optimize Young's modulus, the augmented term and Lagrange term play an essential role in affecting material distribution in the first iteration. When r_p and λ_{ij} are not set properly, even if we use a warm start close to the optimal solution, these two terms may still be prone to derivation from the warm start. As a result, if this derivation causes the current guess too far away from the optimal solution, it will make the iterative process difficult to converge to the solution.

4 EXPERIMENTS

In this section, we conduct experiments to evaluate the performance of the proposed method. We implement the method using C++ and run all the experiments on a computer with Intel(R) Xeon(R) W-2133 and 32GB RAM. Ceres is used for unconstrained nonlinear conjugate gradient or L-BFGS optimization. L-BFGS-B [31] and SNOPT [8] are used for constrained optimization. Intel Pardiso [25] and Eigen lib [10] are used for solving the sparse linear system. Also, a GUI is developed for interactive editing of user's inputs.

To facilitate the evaluation, we propose the following way to generate the ground truth values in the experiments: Given an input hexahedral mesh model, we first assign homogeneous or heterogeneous distribution of Young's modulus to the model. Then we select some vertices from the hexahedral model, apply some forces at these vertices, and also choose a few vertices to be fixed. After that, we perform finite element analysis (FEA) using Vega FEM lib [1] to compute the deformation of the model. The computed deformation (i.e., vertex displacements) and the assigned Young's modulus values serve as the ground truth.

In testing, we take as input the applied forces and the corresponding computed displacements from FEA at the selected vertices, together with the set of fixed vertices. Our method then computes the Young's modulus for each hexahedral element and the deformation displacement for each vertex of the model. The evaluation can be performed by examining the difference of the deformation displacements generated by our method and the ground truth (i.e., generated by the FEA), and as well the difference of the Young's modulus distributions generated by our method and the ground truth. Also in our experiments, we adopt constraints of Young's modulus: $0 \leq \mathrm{E} \leq 10$.

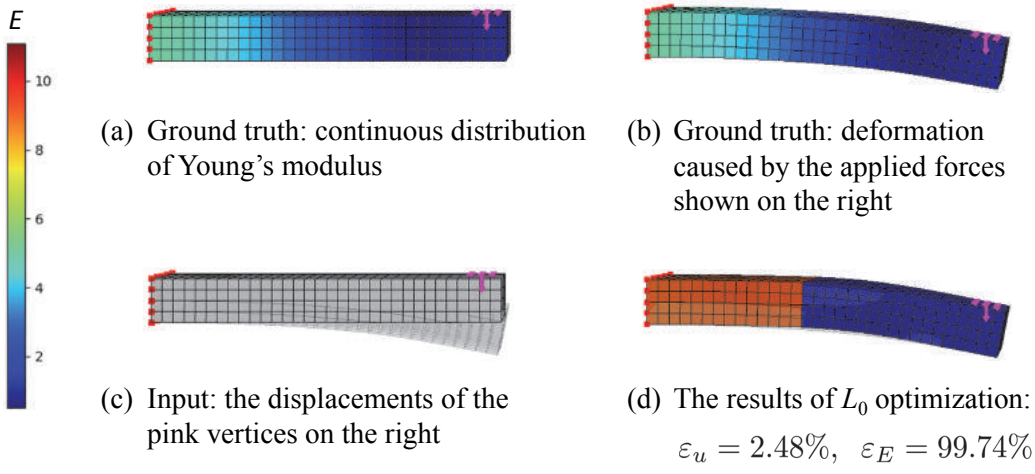

Figure 1: L_0-optimization on a bar model. (a): the ground true of Young's modulus; (b): the deformation caused by applying a force to a few selected vertices shown in pink, which also serves as the ground truth; (c): the displacements at the pink vertices computed in (b) are used as input in testing; (d): the result of our L_0-optimization.

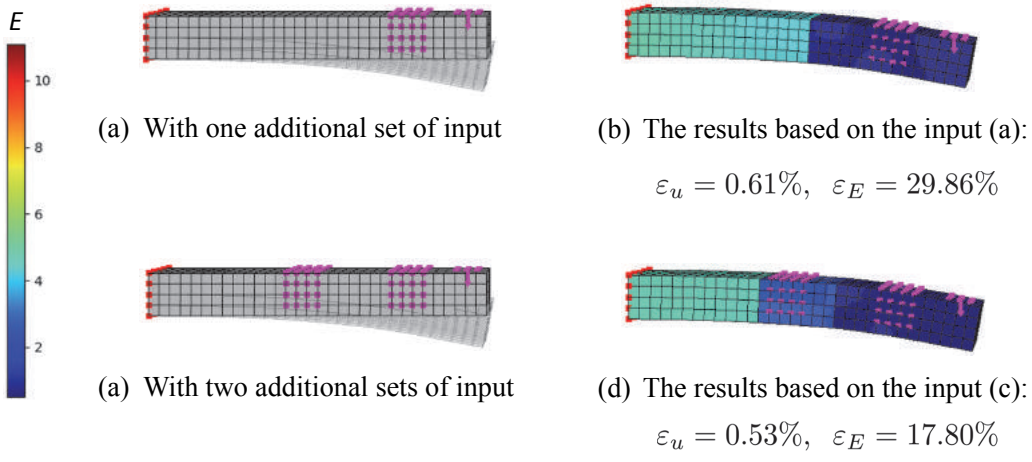

Figure 2: More inputs are added to the example in Fig.1. (a): displacement input at 1 additional set of vertices; (b): the results of the optimization based on the input of (a); (c): displacement input at 2 additional sets of vertices; (d): the results of the optimization based on the input of (c).

Our first example is a simple bar model consisting of 825 vertices and 512 hexahedra. The size of the bar model is 5 cm $\times 5$ cm $\times 50$ cm. We fix the left side of the bar, which is highlighted in red. To generate the ground truth for evaluation, we let the Young's modulus distribution of the bar change linearly from 6e7 Pa on the left to 2e7 Pa in the middle, and to 0.5e7 Pa on the right. The color on the bar in Fig.1(a) depicts

the Young's modulus distribution. A force is applied to a selected set of vertices on the right, which is shown in pink. This force leads to a deformation of the bar, which is computed by FEA based simulation and shown in Fig.1(b). Both Figs.1(a) and (b) then serve as the ground truth in our experiment.

Next we take the force and the displacement at the selected (pink) vertices as input (see Fig.1(c)) and run our optimization algorithm to compute the Young's modulus distribution over the bar model. We aim to find a distribution such that the resulting displacements at the selected pink vertices match the ground truth (i.e., the results obtained from the FEA). In our algorithm, the optimization process stops when the primal and dual residuals are sufficiently small. Following Boyd et al. [6], the residuals are chosen using an absolute and relative criterion. We set absolute tolerance as 1e-2 and relative tolerance as 1e-3. If the optimizer cannot achieve these residual tolerances, we stop at the iteration that follows a long plateau where our optimizer can no longer reduce the primal and dual residuals. In this example, our algorithm takes about 1049 seconds to give a solution that is shown in Fig.1(d). To evaluate the results, we use relative displacement error ε_u and the relative difference ε_E of Young's modulus values, which are defined as follows:

$$\varepsilon_u = \max_{\{v\text{: the selected vertices}\}} \frac{\|SK^{-1}(E)f - \bar{u}\|}{\|\bar{u}\|}, \tag{15}$$

$$\varepsilon_E = \max_{\{e\text{: all elements}\}} \frac{\|E - \bar{E}\|}{\|\bar{E}\|} \tag{16}$$

where $E, \bar{E}$ are the computed and ground truth Young's modulus values, respectively. It can be seen from Fig.1(d) that our L_0-optimization on the bar model can yield sparse Young's modulus distribution and achieve the desired displacements with $\varepsilon_u = 2.48\%$. Meanwhile, the computed Young's modulus distribution is very different from the ground truth with $\varepsilon_E = 99.74\%$. This is actually not a surprise. Since our target is to match the displacements only at a few selected vertices, there exist many material distributions that can achieve this requirement.

It can be expected that when more pairs of forces and displacements are added as input to describe the desired deformation behavior, the Young's modulus obtained in our solution may be closer to the ground truth. This is demonstrated in Fig.2, where we add more inputs. Specifically, we select additional sets of vertices, and use the displacements at these vertices computed by FEA earlier as input. We also let the force at these vertices be zero. Then our optimization process gives improved solutions. Both relative displacement errors and relative Young's modulus difference with the ground truth become smaller. That is, $\varepsilon_u = 0.61\%, \varepsilon_E = 29.86\%$ for the first case with one additional set of input, and $\varepsilon_u = 0.53\%, \varepsilon_E = 17.80\%$ for the second case with two additional sets of input. The computational time is about 1748 seconds and 1179 seconds for the first and second cases, respectively.

The second example is a human sculpture model with 13886 vertices and 11948 hexahedra (see Fig.3). Thus the number of variables is much larger than that of Example 1. We fix the vertices at the bottom. We also assign Young's modulus values of 3e7 Pa, 1e7 Pa, and 5e7 Pa to the vertices from top to bottom (see Fig.3(a)). Different from Example 1, this distribution is discrete. Next we introduce a force applied to the pink vertices located in the forehead of the model, which leads to a deformation as shown in Fig.3(b). Both Figs.3(a) and (b) serve as the ground truth. In Fig.3(c), we take the displacements at the pink vertices as input in testing. The results of our optimization process are shown in Fig.3(d) where $\varepsilon_u = 1.97\%, \varepsilon_E = 56.47\%$. Similarly, we can add more sets of vertices for input, which improve both the accuracy of the deformation and the closeness of the computed Young's modulus to the ground truth, as demonstrated in Fig.4.

Two more examples with the pig and bunny models are given in Figs.5 and 6. The vertices of the models at the bottom are fixed. The applied forces and the target displacement are specified at pink vertices as shown in Figs.5(a) and 6(a), which are the inputs to our algorithm. For each model, two outputs are presented, which correspond to different setup of parameter values of μ and r_p in the process of ADMM. In fact, the ratio $\frac{\mu}{r_p}$ affects the effect of L_0 regularization. The larger the ratio is, the more the algorithm moves $P_{i,j}$ towards

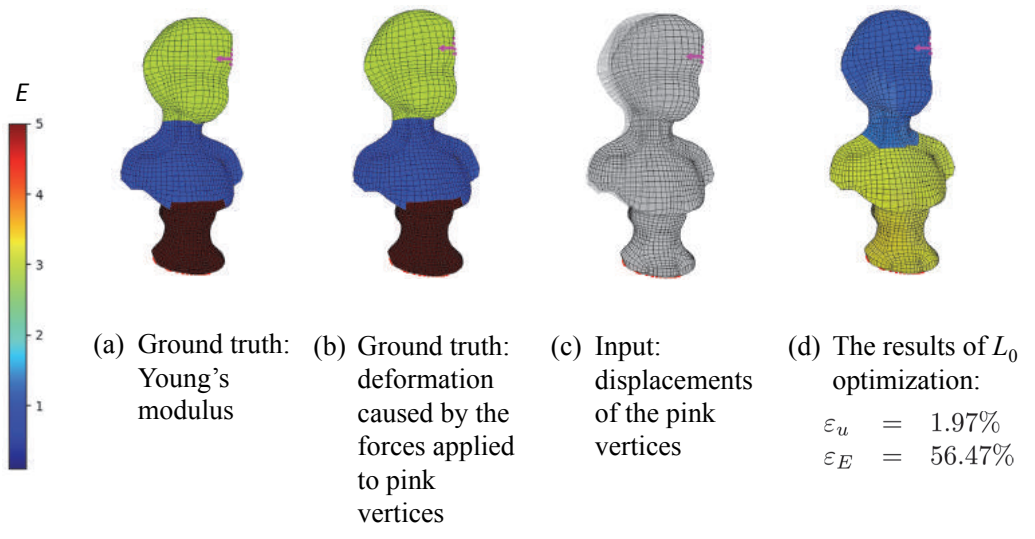

Figure 3: L_0-optimization on a human sculpture model. (a): the ground true of Young's modulus; (b): the deformation caused by applying a force to the pink vertices; (c): the displacements at the pink vertices computed in (b) are used as input in testing; (d): the result of our L_0-optimization.

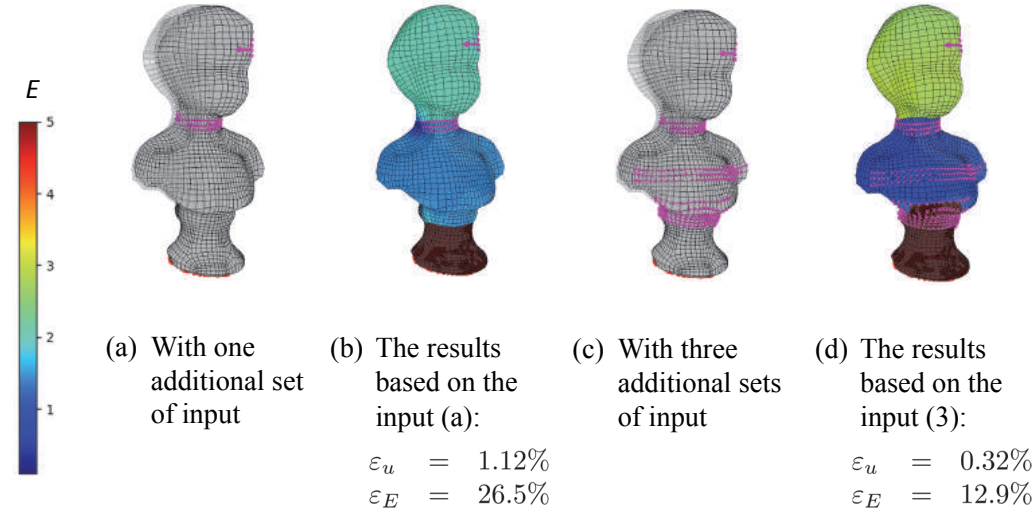

Figure 4: More inputs are added to the example in Fig.3. (a): displacement input at 1 additional set of vertices; (b): the results of the optimization based on the input of (a); (c): displacement input at 3 additional sets of vertices; (d): the results of the optimization based on the input of (c).

0. Note that $P_{i,j} = 0$ means $E_i - E_j = 0$, which makes two adjacent elements i and j have the same Young's modulus. Thus the ratio $\frac{\mu}{r_p}$ can be used to tune the effect of L_0 regularization. In Figs.5(b) and 6(b), the ratio is smaller than that in Figs.5(c) and 6(c). As a consequence, the results in Figs.5(c) and 6(c) appear to

be sparser those in Figs.5(b) and 6(b).

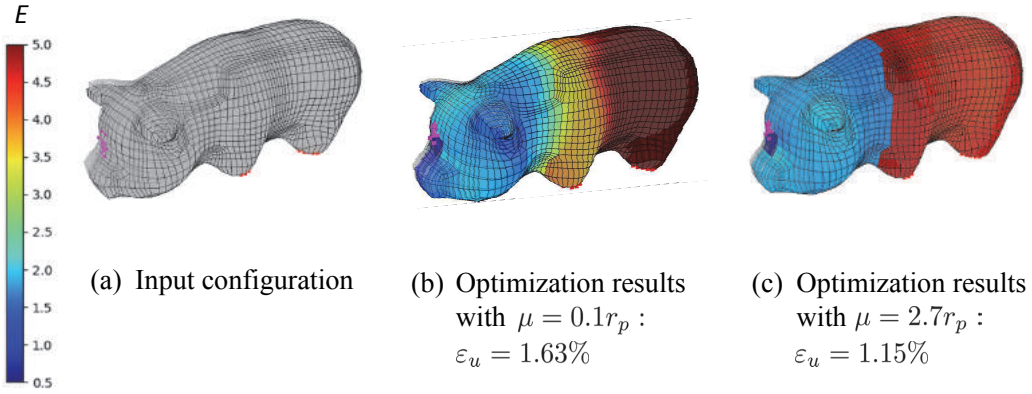

Figure 5: L_0-optimization on a pig model. (a): input configuration; (b): the result of our L_0-optimization with $\mu = 0.1r_p$; (c): the result of our L_0-optimization with $\mu = 2.7r_p$.

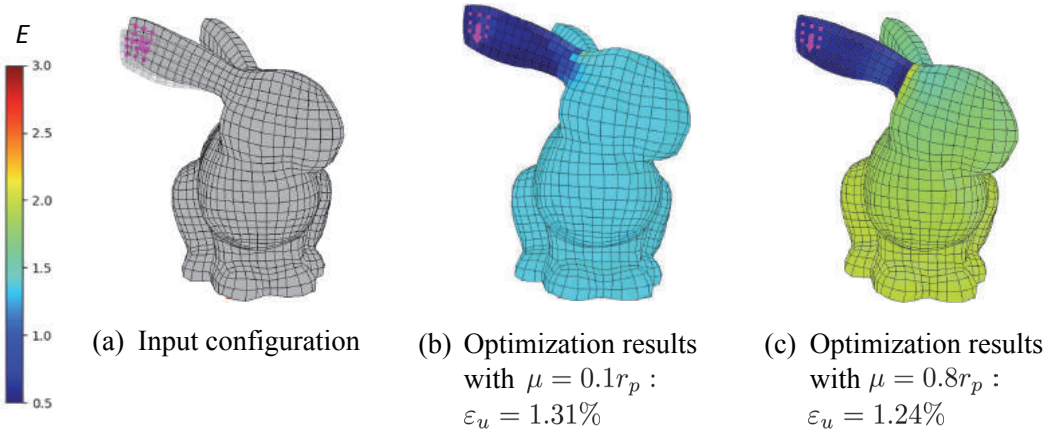

Figure 6: L_0-optimization on a bunny model. (a): input configuration; (b): the result of our L_0-optimization with $\mu = 0.1r_p$; (c): the result of our L_0-optimization with $\mu = 0.8r_p$.

The statistics of the models and the optimization process in our experiments are summarized in Table 1. The tested models cover small to relatively large mesh density. Our optimization algorithm can deliver the results that achieve the desired deformation behavior. It is also noticed that the our algorithm does require certain amount of computational time.

5 CONCLUSIONS

This paper has presented a numerical method for calculating Young's modulus distribution over a hexahedral mesh to achieve desired deformation. The method is based on L_0-optimization that can be regarded as a new

Table 1: Statistics of the models and optimization process in Figs.1-6.

Models	Number of vertices	Number of hexahedra	ε_u	Time (seconds)
Bar (Fig.1(d))	825	512	2.48%	1049
Bar (Fig.2(b))			0.61%	1748
Bar (Fig.2(d))			0.53%	1179
Human sculpture (Fig.3(d))	13886	11948	1.97%	12952.8
Human sculpture (Fig.4(b))			1.12%	26025.3
Human sculpture (Fig.4(d))			0.32%	18051.6
Pig (Fig.5(c))	12887	11218	1.15%	10044.9
Bunny (Fig.6(c))	8756	7455	1.24%	8525.5

way to design underlying material distribution to achieve the desired deformation while encouraging sparse material distribution. It can help to explore design variations and also satisfy the sparsity requirement, which may find applications such as 3D printing. Also, we implement the proposed L_0-optimization by introducing auxiliary variables in a carefully designed objective function and performing numerical iteration in a special order. The effectiveness of the algorithm is demonstrated in several examples.

Our method has several limitations. First, the overall computational time is general long due to L_0 regularization. The subspace method proposed by Xu et al. [28] might help in speeding up. Incorporating subspace into the L_0 optimization is worthy of investigation. Second, our method contains some additional parameters like μ, r_p in the optimization. How to automatically choose good values for these parameters is non-trivial. Third, we only optimize Young's modulus in our problem. Poisson's ratio has an effect on deformation [21, 17], which might be considered for future work.

ACKNOWLEDGEMENTS

This work is supported by the Ministry of Education, Singapore, under its MoE Tier-2 Grant (2017-T2-1-076).

ORCID

Haoxiang Li, http://orcid.org/0000-0002-7002-9007
Jianmin Zheng, http://orcid.org/0000-0002-5062-6226

REFERENCES

[1] Barbič, J.; Sin, F.S.; Schroeder, D.: Vega FEM Library, 2012. Http://www.jernejbarbic.com/vega.

[2] Becker, M.; Teschner, M.: Robust and efficient estimation of elasticity parameters using the linear finite element method. In SimVis, 15–28, 2007.

[3] Besl, P.J.; McKay, N.D.: Method for registration of 3-d shapes. In Sensor fusion IV: control paradigms and data structures, vol. 1611, 586–606. International Society for Optics and Photonics, 1992. http://doi.org/10.1117/12.57955.

[4] Bickel, B.; Bächer, M.; Otaduy, M.A.; Lee, H.R.; Pfister, H.; Gross, M.; Matusik, W.: Design and fabrication of materials with desired deformation behavior. ACM Transactions on Graphics (TOG), 29(4), 1–10, 2010. http://doi.org/10.1145/1833351.1778800.

[5] Bickel, B.; Bächer, M.; Otaduy, M.A.; Matusik, W.; Pfister, H.; Gross, M.: Capture and modeling of non-linear heterogeneous soft tissue. ACM transactions on graphics (TOG), 28(3), 1–9, 2009. http://doi.org/10.1145/1531326.1531395.

[6] Boyd, S.; Parikh, N.; Chu, E.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Now Publishers Inc, 2011. http://doi.org/10.1561/2200000016.

[7] Chen, X.; Zheng, C.; Xu, W.; Zhou, K.: An asymptotic numerical method for inverse elastic shape design. ACM Transactions on Graphics (TOG), 33(4), 1–11, 2014. http://doi.org/10.1145/2601097.2601189.

[8] Gill, P.E.; Murray, W.; Saunders, M.A.: Snopt: An sqp algorithm for large-scale constrained optimization. SIAM review, 47(1), 99–131, 2005. http://doi.org/10.2307/20453604.

[9] Gloria, A.: Numerical homogenization: survey, new results, and perspectives. In ESAIM: Proceedings, vol. 37, 50–116. EDP Sciences, 2012. http://doi.org/10.1051/proc/201237002.

[10] Guennebaud, G.; Jacob, B.; et al.: Eigen v3. http://eigen.tuxfamily.org, 2010.

[11] Hašan, M.; Fuchs, M.; Matusik, W.; Pfister, H.; Rusinkiewicz, S.: Physical reproduction of materials with specified subsurface scattering. In ACM SIGGRAPH 2010 papers, 1–10, 2010. http://doi.org/10.1145/1833351.1778798.

[12] He, L.; Schaefer, S.: Mesh denoising via L0 minimization. ACM Transactions on Graphics (TOG), 32(4), 1–8, 2013. http://doi.org/10.1145/2461912.2461965.

[13] Hoffmann, C.M.: Geometric and Solid Modeling: An Introduction. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1989. ISBN 1558600671.

[14] Kharevych, L.; Mullen, P.; Owhadi, H.; Desbrun, M.: Numerical coarsening of inhomogeneous elastic materials. ACM Transactions on graphics (TOG), 28(3), 1–8, 2009. http://doi.org/10.1145/1576246.1531357.

[15] Kim, N.H.: Introduction to nonlinear finite element analysis. Springer Science & Business Media, 2014. http://doi.org/10.1007/978-1-4419-1746-1.

[16] Leung, Y.; Kwok, T.; Mao, H.; Chen, Y.: Digital material design using tensor-based error diffusion for additive manufacturing. Comput. Aided Des., 114, 224–235, 2019. http://doi.org/10.1016/j.cad.2019.05.031.

[17] Long, K.; Yuan, P.F.; Xu, S.; Xie, Y.M.: Concurrent topological design of composite structures and materials containing multiple phases of distinct poisson's ratios. Engineering Optimization, 50(4), 599–614, 2018. http://doi.org/10.1080/0305215X.2017.1337757.

[18] Morovič, P.; Morovič, J.; Tastl, I.; Gottwals, M.; Dispoto, G.: Co-optimization of color and mechanical properties by volumetric voxel control. Structural and Multidisciplinary Optimization, 60(3), 895–908, 2019. http://doi.org/10.1007/s00158-019-02240-8.

[19] Nealen, A.; Müller, M.; Keiser, R.; Boxerman, E.; Carlson, M.: Physically based deformable models in computer graphics. In Computer graphics forum, vol. 25, 809–836. Wiley Online Library, 2006. http://doi.org/10.1111/j.1467-8659.2006.01000.x.

[20] Nguyen, R.M.; Brown, M.S.: Fast and effective l0 gradient minimization by region fusion. In Proceedings of the IEEE international conference on computer vision, 208–216, 2015. http://doi.org/10.1109/ICCV.2015.32.

[21] Shufrin, I.; Pasternak, E.; Dyskin, A.V.: Hybrid materials with negative poisson's ratio inclusions. International Journal of Engineering Science, 89, 100–120, 2015. http://doi.org/10.1016/j.ijengsci.2014.12.006.

[22] Sifakis, E.; Barbic, J.: Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction. In Acm siggraph 2012 courses, 1–50, 2012. http://doi.org/10.1145/2343483.2343501.

[23] Skouras, M.; Thomaszewski, B.; Coros, S.; Bickel, B.; Gross, M.: Computational design of actuated deformable characters. ACM Transactions on Graphics (TOG), 32(4), 1–10, 2013. http://doi.org/10.1145/2461912.2461979.

[24] Terzopoulos, D.; Platt, J.; Barr, A.; Fleischer, K.: Elastically deformable models. In Proceedings of the 14th annual conference on Computer graphics and interactive techniques, 205–214, 1987.

[25] Wang, E.; Zhang, Q.; Shen, B.; Zhang, G.; Lu, X.; Wu, Q.; Wang, Y.: Intel math kernel library. In High-Performance Computing on the Intel® Xeon Phi, 167–188. Springer, 2014. http://doi.org/10.1007/978-3-319-06486-4_7.

[26] Weiss, S.; Maier, R.; Cremers, D.; Westermann, R.; Thuerey, N.: Correspondence-free material reconstruction using sparse surface constraints. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4686–4695, 2020. http://doi.org/10.1109/CVPR42600.2020.00474.

[27] Xi, S.; Chen, H.; Wu, B.I.; Kong, J.A.: One-directional perfect cloak created with homogeneous material. IEEE Microwave and Wireless Components Letters, 19(3), 131–133, 2009. http://doi.org/10.1109/LMWC.2009.2013677.

[28] Xu, H.; Li, Y.; Chen, Y.; Barbič, J.: Interactive material design using model reduction. ACM Transactions on Graphics (TOG), 34(2), 1–14, 2015. http://doi.org/10.1145/2699648.

[29] Xu, L.; Lu, C.; Xu, Y.; Jia, J.: Image smoothing via l0 gradient minimization. In Proceedings of the 2011 SIGGRAPH Asia Conference, 1–12, 2011. http://doi.org/10.1145/2070752.2024208.

[30] Zhang, J.; Zhao, D.; Gao, W.: Group-based sparse representation for image restoration. IEEE Transactions on Image Processing, 23(8), 3336–3351, 2014. http://doi.org/10.1109/TIP.2014.2323127.

[31] Zhu, C.; Byrd, R.H.; Lu, P.; Nocedal, J.: Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. ACM Transactions on Mathematical Software (TOMS), 23(4), 550–560, 1997. http://doi.org/10.1145/279232.279236.