

# A generalized DCT compression based density method for topology optimization of 2D and 3D continua

Pingzhang Zhou<sup>a,\*</sup>, Jianbin Du<sup>b,\*</sup>, Zhenhua Lü<sup>a</sup>

<sup>a</sup> *Department of Automotive Engineering, Tsinghua University, Beijing, PR China*

<sup>b</sup> *School of Aerospace Engineering, Tsinghua University, Beijing, PR China*

Received 20 August 2017; received in revised form 26 November 2017; accepted 29 January 2018

Available online 5 February 2018

## Highlights

- A novel efficient topology optimization method based on image compression is proposed.
- The number of design variables can be phenomenally reduced.
- Exact bounds of design variables and analytical sensitivity information are derived.
- No additional filter is needed.
- No sharp corners are presented in the final optimized topology.

## Abstract

In this paper, a novel topology optimization method based on discrete cosine transform (DCT) and density interpolation is proposed for layout designs of 2D and 3D continua. As one of the most frequently used transforms in digital image compression, the DCT may significantly reduce the number of design variables in density-based topology optimization, and can hereby improve the efficiency of solving the topology optimization problems to a great extent. This way the DCT compression based density method (DCDM) could be quite attractive in the topology optimization of large-scale engineering structures where a huge number of design variables may present. Effectiveness and efficiency of the proposed method is demonstrated with several 2D and 3D examples including both mechanical and heat conduction problems. Through these examples, some interesting features of DCDM are revealed and discussed. Since high frequency components are inherently filtered in DCDM, there is no need to introduce additional density filter or sensitivity filter in the present model. It is shown by numerical examples that there is no sharp corners present in the final optimized layout obtained by DCDM, which is beneficial when considering the stress of structures.

© 2018 Elsevier B.V. All rights reserved.

**Keywords:** Topology optimization; Discrete cosine transform (DCT); Digital image compression; Density method

\* Corresponding authors.

E-mail addresses: [zp13@mails.tsinghua.edu.cn](mailto:zp13@mails.tsinghua.edu.cn) (P. Zhou), [dujb@tsinghua.edu.cn](mailto:dujb@tsinghua.edu.cn) (J. Du).

## 1. Introduction

Recent years have witnessed the popularity of topology optimization methods in various structural design problems. An apparent evidence can be the implementations of topology optimization module in more and more commercial CAE software, such as Nastran, ABAQUS, ANSYS, COMSOL, Altair Optistruct, TOSCA, etc.

After about 30 years' development since the pioneering work by Bense and Kikuchi [1], now there are many methods available for topology optimization problems such as homogenization method [1], density method [2–4], evolution method (ESO/BESO) [5,6], level-set method [7] and the newly proposed moving morphable components method [8], etc. However, until now the common sense among this community is that the density method is still the most popular and widely-used approach. In traditional density method, the elemental pseudo density is taken as the design variable. So the number of design variables is the same as the number of elements in the FEA model. In general, to acquire clear and crisp layout it requires an extremely fine mesh, so the computational burden is heavy in both the FEA analysis and the optimization process.

The requirement is especially troublesome when the design domain is in 3D space, which is exactly the case in engineering practice. Indeed it should be admitted that the application of topology optimization to the design of large engineering structures such as bridges, cars or airplanes is currently hindered because the number of design variables is so large that the optimization is both intractable and costly for even the fastest computers available today [9].

Since the topology optimization requires finer mesh than FEA analysis, a multi-resolution implementation can be used where different levels of discretization are used for the displacement mesh, design variable mesh and density mesh [10]. However the multi-resolution implementation is by no means concise and would destroy the beauty of simplicity of the original density method.

Recently, Guo and his co-workers [8,11–13] proposed the moving morphable components method (MMC) where the layout of structures is explicitly expressed using topology description function (TDF). Due to the explicit expression of structural topology, the variables needed are greatly reduced so that the optimization procedure when using MMC method can be quite efficient. The success of MMC method implies that to describe the layout of a structure, there is no need to use a huge number of design variables. Similar to the basic idea of MMC method that only a few number of variables are needed to achieve the topology optimization, in this paper we try to carry out highly efficient topology optimization method which is still based on the density method and inherits the beauty of simplicity as well.

Density-based topology optimization shares many similarities with digital image processing procedures. Digital images are nowadays transferred innumerable times at any given instant through the internet, so certainly the transfer process should be highly efficient. A digital image of size  $1024 \times 1024$  is quite common in the Internet, but the topology optimization of a 2D structure with  $1024 \times 1024$  meshes is still time-consuming until now. Of course the inefficiency of the latter owes mainly to the FEA analysis and numerical optimization process, but there is something apart from that: the image compression, which is the art and science of reducing the amount of data required to represent an image [14]. In fact, the compression ratio (CR) can easily exceed 10 for most images by using image compression techniques.

Image compression methods can be categorized as lossless methods such as Huffman coding, and lossy methods such as transform coding, Chroma sub-sampling, etc. Discrete cosine transform (DCT)-based methods belong to the transform coding methods which has reached a level of maturity and is the basis of many international image (JPEG) and audio (MP3) compression standards. More details on the origin and development of DCT and its wide applications can be found in [15]. In this paper we focus on using DCT-based image compression to reduce the number of design variables in density-based topology optimization for both 2D and 3D continuum.

By the way, we mention that the efficiency of a topology optimization process is influenced by two factors, i.e. (1) the number of design variables in numerical optimization process and (2) the number of dofs in FEA process. In this paper only the first factor is handled by reducing the number of design variables using image compression techniques. In MMC/MMV, the number of dofs can also be reduced by ignoring the elemental stiffness matrices of the void elements [16,17].

The rest of this paper is arranged as follows. In Section 2 the one-dimensional DCT is introduced in the first place, upon which the multi-dimensional DCT can be easily generalized later in Section 3. The basic principles for data compression are also established using 1D DCT in Section 2 and will be precisely followed afterwards. In Section 4 the framework of topology optimization using DCDM is formally established and carried out. All the technical details including sensitivity analysis, choosing initial solution, and determining bounds for design variables are elaborated in

Section 4 as well. Abundant numerical examples, in both 2D and 3D space covering mechanical and heat conduction problems, are shown in Section 5 to demonstrate the effectiveness and efficiency of the proposed method. In Section 6 some discussions are given to illustrate the relation between the number of nonzero frequency components in our method and the filter radius in the traditional density method. Also present in Section 6 is the discussion on what we lose by using the DCT-based compression, which is categorized as a lossy method in image compression. Finally some conclusions are drawn in Section 7. In the Appendix the intricate derivation of the matrix form of 3D DCT is given as a reference.

Throughout this paper matrices are always denoted by bold letters such as  $\mathbf{g}$ ,  $\mathbf{G}$ , or equivalently by  $[g_{xyz}]$ ,  $[G_{uvw}]$ ; indices are always denoted by lowercase such as  $u, v, w, x, y, z, i, j, k$ , etc.;  $X, Y, Z$  represent global  $X, Y$  and  $Z$  directions, respectively. We count a sequence beginning with 0 to follow the convention in digital image processing.

## 2. One-dimensional DCT and basic principles for data compression

In this section the one-dimensional DCT, together with its good properties, is briefly introduced first, upon which the basic principles for data compression can be shown. As can be seen in later sections, 2D and 3D DCT can simply be achieved by two and three times of 1D DCT, respectively. So the principles established in this section are vital to understanding the success of DCT-based compression in topology optimization.

### 2.1. One-dimensional DCT

Give a sequence  $\{g_0, g_1, \dots, g_{N-1}\}$  in spatial domain, the DCT transform maps it into the sequence  $\{G_0, G_1, \dots, G_{N-1}\}$  in frequency domain<sup>1</sup> by:

$$G_u = \sum_x \left\{ \alpha_u g_x \cos \frac{\pi(2x+1)u}{2N} \right\}, \quad (1)$$

and the inverse discrete transform (IDCT) maps the frequency domain back into spatial domain:

$$g_x = \sum_u \left\{ \alpha_u G_u \cos \frac{\pi(2x+1)u}{2N} \right\}, \quad (2)$$

where  $x, u \in \{0, 1, \dots, N-1\}$ ; the constant  $\alpha_u$  is given by:

$$\alpha_u = \sqrt{\frac{2 - \delta_u}{N}}, \quad (3)$$

where  $\delta_u = 1$  iff  $u = 0$ , otherwise  $\delta_u = 0$ .

It is advantageous to write the 1D DCT in compact form:

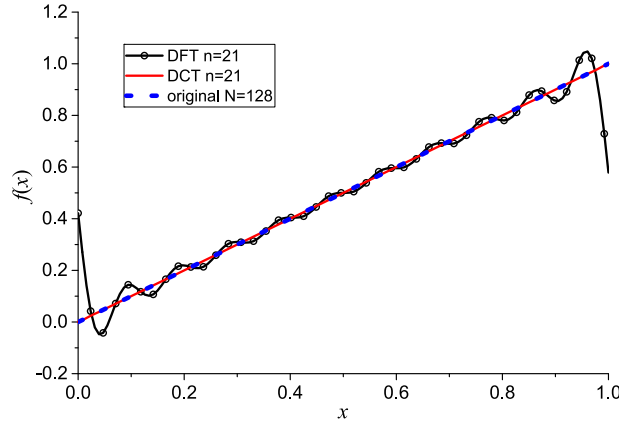
$$\mathbf{G} = \mathbf{D}^N \mathbf{g}, \quad G_u = \sum_x D_{ux}^N g_x, \quad (4)$$

where  $\mathbf{g} = [g_0, g_1, \dots, g_{N-1}]^T$  and  $\mathbf{G} = [G_0, G_1, \dots, G_{N-1}]^T$ .  $\mathbf{D}^N$  is called the transformation matrix (of dimension  $N$ ) and can be explicitly given by<sup>2</sup>:

$$\mathbf{D}^N = \sqrt{\frac{2}{N}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ \cos \frac{\pi}{2N} & \cdots & \cos \frac{2N-1}{2N} \pi \\ \vdots & \ddots & \vdots \\ \cos \frac{N-1}{2N} \pi & \cdots & \cos \frac{(2N-1)(N-1)}{2N} \pi \end{bmatrix}. \quad (5)$$

<sup>1</sup> Mathematically one can simply call  $\{g_x\}$  the original domain and  $\{G_u\}$  the transformed domain, but in image compression it is beneficial to establish the concepts of spatial domain and frequency domain.

<sup>2</sup> In MATLAB, one can simply obtain  $\mathbf{D}^N$  by using `dctmtx(N)`.



**Fig. 1.** Data compression of a 1D sequence using DFT and DCT. The original data samples  $f(x) = x$ ,  $x \in [0, 1]$  using  $N = 128$  points, i.e.  $\mathbf{g} = [\frac{0}{127}, \frac{1}{127}, \dots, \frac{127}{127}]$ . By using the data compression procedure, DCT and DFT (by replacing  $\mathbf{D}^N$  with the Fourier matrix) reconstruct the original data using only  $n = 21$  nonzero frequency components.

By some simple algebra, it can be demonstrated that  $\mathbf{D}^N$  is orthogonal (i.e.  $(\mathbf{D}^N)^\top \mathbf{D}^N = \mathbf{I}^N$ ). It follows that the 1D IDCT can be given by:

$$\mathbf{g} = (\mathbf{D}^N)^\top \mathbf{G}, \quad g_x = \sum_u D_{ux}^N G_u. \quad (6)$$

## 2.2. Basic principles for data compression

Now the basic mechanism for data compression could be shown using the 1D DCT. Given a 1D sequence  $\mathbf{g} = [g_0, g_1, \dots, g_{N-1}]^\top$ , there is no need to store all the  $N$  numbers in spatial domain. Instead, only quite a few nonzero DCT coefficients are needed to reconstruct the original signal. The procedure works as follows:

- (1) Compute the frequency domain components  $\mathbf{G} = [G_0, G_1, \dots, G_{N-1}]^\top$  using the 1D DCT (4).
- (2) Retain only the low frequency components while discard all high frequency components. This way we obtain  $\tilde{\mathbf{G}} = [G_0, G_1, \dots, G_{n-1}, 0, \dots, 0]^\top$ .  $n \ll N$  is the number of components that will be retained. Notice that only nonzero components need to be stored, the zeros can be padded whenever needed.
- (3) The original sequence in spatial domain can now be approximated using the truncated sequence in frequency domain by 1D IDCT:

$$\mathbf{g} \approx (\mathbf{D}^N)^\top \tilde{\mathbf{G}}. \quad (7)$$

In Fig. 1 an example is shown where the original data is of length  $N = 128$ , and the reconstructed data uses only  $n = 21$  nonzero frequency components. As can be seen, the DCT-based compression perfectly recovers the original data while the discrete Fourier transform (DFT)-based compression behaves poorly.

The fact that Fourier series fails to converge uniformly at discontinuities is called the Gibbs phenomenon. Recall that by using DCT or DFT, the sequence is implicitly extended periodically. From Fig. 2 one can find that for a sequence of length  $N$ , DFT implicitly treats it as a sequence of infinite length with periodicity  $N$ , so if  $g(0)$  is not coincident with  $g(N)$ , the discontinuity occurs.<sup>3</sup> DCT can avoid Gibbs phenomenon since it actually computes the DFT of  $2N$  points by periodical and symmetrical extension. The origin and possible ways to circumvent the Gibbs phenomenon can be found in [18]. Here we comment that some common transforms in mathematical community, such

<sup>3</sup> It would be a little intricate to rigorously explain the Gibbs phenomenon. In fact, originally the Gibbs phenomenon is related to FS (Fourier series) of periodical functions ranging in  $(-\infty, +\infty)$ . In DFT, both the frequency domain and spatial domain are of finite length but is implicitly periodical, so no Gibbs phenomenon occurs. However, if a DFT sequence is truncated (which is exactly the case in data compression), the Gibbs phenomenon occurs.

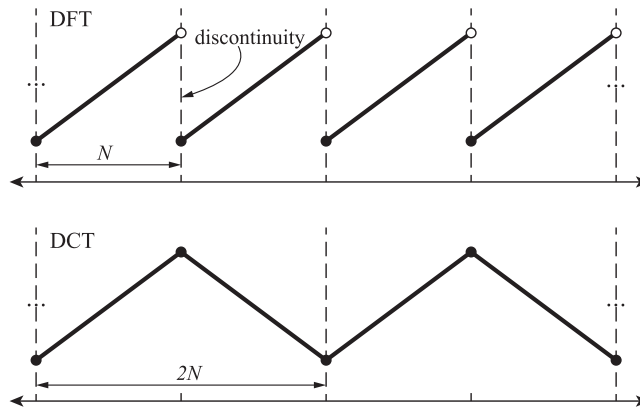


Fig. 2. Illustration of the periodical extension implied by DCT and DFT.

as Fourier, Chebyshev and Legendre polynomials, are all suffered from severe Gibbs phenomenon. Gegenbauer and Jacobi polynomials can find ways to get rid of Gibbs phenomenon [18] but they have very complicated expressions and are inefficient to compute. The algebraic polynomial interpolation is well-known to suffer from Runge phenomenon under equispaced grid so is also inappropriate for data compression.

### 2.3. Good properties of DCT pair

Now it is time to conclude the good properties of DCT to show why it is appropriate for data compression and can be borrowed to be applied in topology optimization:

- (1) It has explicit and friendly analytical expression so that the sensitivity information can be easily derived.
- (2) It is free from Gibbs phenomenon.
- (3) It is free from Runge phenomenon under equispaced grid.
- (4) It has fast computation algorithms,<sup>4</sup> so there is no need to explicitly compute Eqs. (1), (2).
- (5) It is orthogonal and separable. So 2D and 3D DCTs can be implemented through a series of 1D DCT.

All the abovementioned properties, except for the separability, can be seen clearly in 1D DCT. The separability will be demonstrated in 2D and 3D DCTs shortly in the next section.

### 3. Extensions to multi-dimensional DCTs

Density-based topology optimization has many similarities with digital image processing. Many techniques now used in topology optimization, such as the linear [19] and nonlinear [20] density filter, can find their counterparts in digital image processing.

The elemental fictitious density of a 2D rectangular structure can be equivalently seen as a still digital image:

$$\rho = [g_{xy}], \quad (8)$$

where  $x \in \{0, 1, \dots, N_X - 1\}$ ,  $y \in \{0, 1, \dots, N_Y - 1\}$ ,  $g_{xy} \in [0, 1]$ ,  $N_X$  and  $N_Y$  are the number of elements in global  $X$  and  $Y$  directions, respectively.

Similarly in a 3D cubic structure the elemental fictitious density can be seen as either a video or simply a 3D image (e.g. human heart slices) [15]:

$$\rho = [g_{xyz}], \quad (9)$$

where  $x \in \{0, 1, \dots, N_X - 1\}$ ,  $y \in \{0, 1, \dots, N_Y - 1\}$ ,  $z \in \{0, 1, \dots, N_Z - 1\}$ ,  $g_{xyz} \in [0, 1]$ ,  $N_X$ ,  $N_Y$  and  $N_Z$  is the number of elements in global  $X$ ,  $Y$  and  $Z$  directions, respectively.

<sup>4</sup> There are a lot of fast algorithms for DCT. In this paper the *dct*, *dctmtx* or *dct2* in MATLAB is used.

### 3.1. Two-dimensional DCT

Given a 2D image  $[g_{xy}]$  in spatial domain, the corresponding frequency domain components can be given by the 2D DCT as:

$$G_{uv} = \sum_{x,y} \left\{ \alpha_u \alpha_v g_{xy} \cos \frac{\pi(2x+1)u}{2N_X} \cos \frac{\pi(2y+1)v}{2N_Y} \right\}, \quad (10)$$

where  $x, u \in \{0, 1, \dots, N_X - 1\}$ ;  $y, v \in \{0, 1, \dots, N_Y - 1\}$ ;  $N_X$  and  $N_Y$  are the number of pixels in global  $X$  and  $Y$  directions, respectively;  $\alpha_u, \alpha_v$  are the constants:

$$\alpha_u = \sqrt{\frac{2 - \delta_u}{N_X}} \quad \alpha_v = \sqrt{\frac{2 - \delta_v}{N_Y}} \quad (11)$$

where  $\delta_u, \delta_v = 1$  iff  $u, v = 0$ ; otherwise  $\delta_u, \delta_v = 0$ .

Rewrite Eq. (10) as:

$$S_{uy} = \alpha_u \sum_x \left\{ g_{xy} \cos \frac{\pi(2x+1)u}{2N_X} \right\}, \quad (12a)$$

$$G_{uv} = \alpha_v \sum_y \left\{ S_{uy} \cos \frac{\pi(2y+1)v}{2N_Y} \right\}, \quad (12b)$$

where  $x, u \in \{0, 1, \dots, N_X - 1\}$ ;  $y, v \in \{0, 1, \dots, N_Y - 1\}$ .

Eq. (12) reveals the separability of the DCT: the 2D DCT can simply be interpreted as first applying 1D DCT to each column (the summation on  $x$ ) of  $[g_{xy}]$  to obtain the semi-transformed matrix  $[S_{uy}]$ , and then applying 1D DCT to each row (the summation on  $y$ ) of  $[S_{uy}]$  to obtain  $[G_{uv}]$ .

The compact form of the 2D DCT can now be given by:

$$\mathbf{G} = \mathbf{D}^{N_X} \mathbf{g} (\mathbf{D}^{N_Y})^\top, \quad G_{uv} = \sum_{x,y} D_{ux}^{N_X} D_{vy}^{N_Y} g_{xy}, \quad (13)$$

where  $\mathbf{D}^{N_X}$  and  $\mathbf{D}^{N_Y}$  can be obtained by replacing  $N$  with  $N_X$  and  $N_Y$  in Eq. (5), respectively.

Recall that  $\mathbf{D}^N$  is orthogonal, so the 2D IDCT can be given by:

$$\mathbf{g} = (\mathbf{D}^{N_X})^\top \mathbf{G} \mathbf{D}^{N_Y}, \quad g_{xy} = \sum_{u,v} D_{ux}^{N_X} D_{vy}^{N_Y} G_{uv}. \quad (14)$$

Fig. 3 vividly shows the compression-restoration process of a 2D image. Note that Fig. 3(c) is obtained by the 2D IDCT of (d), so the actual compression–restoration process is (a)–(b)–(d)–(c). The values in the color bars in Fig. 3 are log-scale. From Fig. 3 one can clearly find that most energy of an image is distributed in the low-frequency components. The compression–restoration techniques work perfectly in this example and only minor differences can be detected if (a) and (c) are zoomed and compared.

### 3.2. Three-dimensional DCT

As illustrated in Fig. 4, the index numbering may be not that trivial in a 3D matrix. Specifically, a *column* denotes an array having the same  $x$  and  $z$  values; a *row* denotes an array having the same  $y$  and  $z$  values; a *tube* denotes an array having the same  $x$  and  $y$  values. The term *frontal slice* refers to the arrays having the same  $z$  value. Please refer to [21] for more explanations on third order tensor.

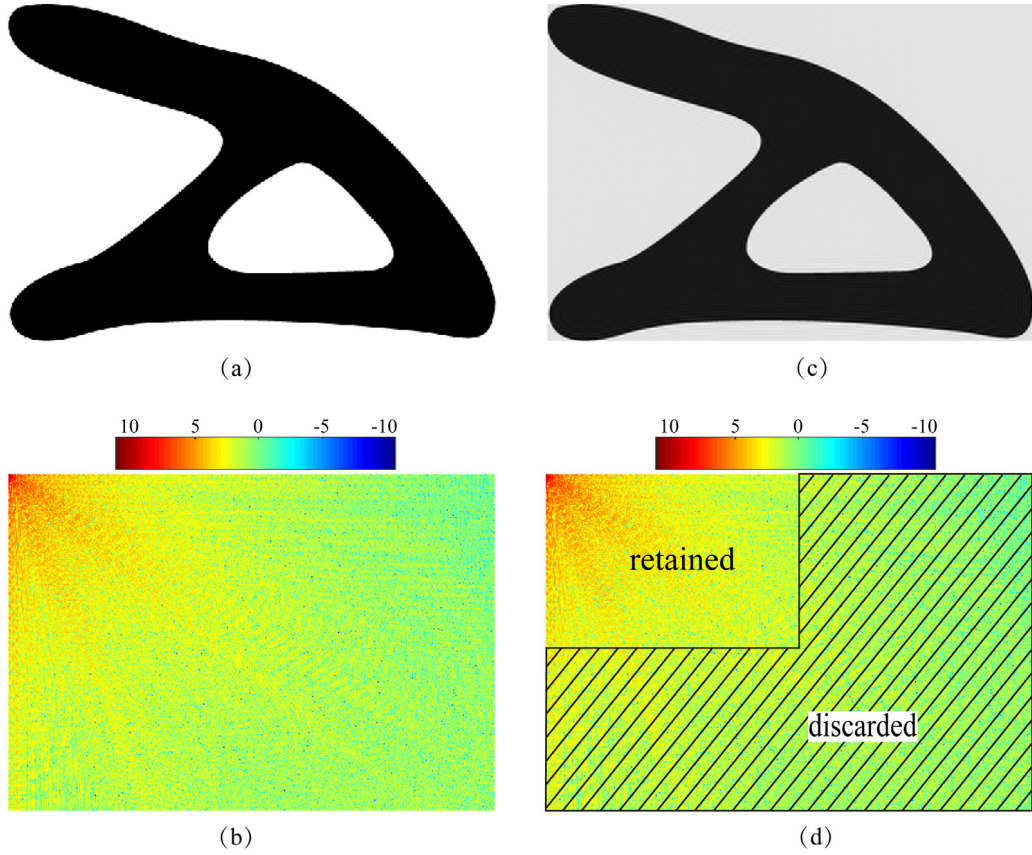
Based on the separability of DCT, now we can directly give the compact form of 3D DCT by generalizing Eq. (13):

$$G_{uvw} = \sum_{x,y,z} D_{ux}^{N_X} D_{vy}^{N_Y} D_{wz}^{N_Z} g_{xyz}, \quad (15)$$

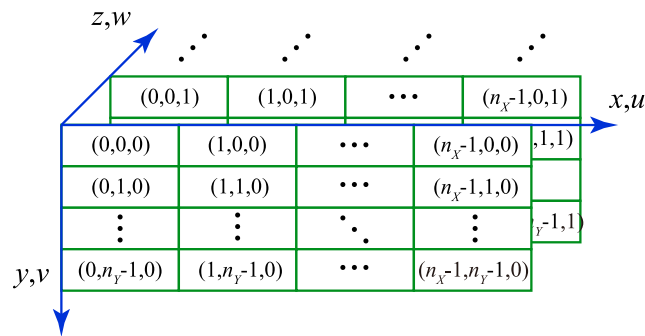
and 3D IDCT by generalizing Eq. (14):

$$g_{xyz} = \sum_{u,v,w} D_{ux}^{N_X} D_{vy}^{N_Y} D_{wz}^{N_Z} G_{uvw}, \quad (16)$$





**Fig. 3.** The compression–restoration process of a 2D image.  $N_X = 591$ ,  $N_Y = 409$ ,  $n_X = 300$ ,  $n_Y = 200$ . (a) The original image. (b) Contour of the DCT coefficients of the original image. (c) The compressed image. (d) Discard all high-frequency components of the DCT coefficients. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



**Fig. 4.** The index numbering in 3D matrix.

where  $x, u \in \{0, 1, \dots, N_X - 1\}$ ;  $y, v \in \{0, 1, \dots, N_Y - 1\}$ ;  $z, w \in \{0, 1, \dots, N_Z - 1\}$ ;  $\mathbf{D}^{N_X}$ ,  $\mathbf{D}^{N_Y}$  and  $\mathbf{D}^{N_Z}$  can be obtained by replacing  $N$  with  $N_X$ ,  $N_Y$  and  $N_Z$  in Eq. (5), respectively.

To the best of our knowledge, there is no matrix form available for the 3D DCT in existing literature. By using some tricks from tensor analysis, it is possible to find the matrix form for Eqs. (15), (16). However, the derivation

process is somewhat tricky and complicated. So we move the detailed derivation process to the [Appendix](#) section and directly give the matrix form for 3D DCT:

$$\mathbf{G} = \left( \mathbf{D}^{N_Z} \left( \mathbf{D}^{N_Y} \left( \mathbf{D}^{N_X} \mathbf{g} \right)^{\top_{[231]}} \right)^{\top_{[231]}} \right)^{\top_{[231]}}, \quad (17)$$

and 3D IDCT:

$$\mathbf{g} = \left( \left( \mathbf{D}^{N_Z} \right)^{\top} \left( \left( \mathbf{D}^{N_Y} \right)^{\top} \left( \left( \mathbf{D}^{N_X} \right)^{\top} \mathbf{G} \right)^{\top_{[231]}} \right)^{\top_{[231]}} \right)^{\top_{[231]}}. \quad (18)$$

The correctness of Eqs. (17), (18) can be easily verified by the indicial notation of these equations using the knowledge from tensor analysis. Also, one can find that the matrix form of 1D and 2D DCTs and IDCTs can be degenerated from Eqs. (17), (18).

Recall that the basic principles we established in 2D DCT still hold: one can simply apply 1D DCT sequentially to each column, row and tube of a 3D image  $[g_{xyz}]$  to acquire its frequency components  $[G_{uvw}]$ , and similarly for 3D IDCT.

#### 4. Topology optimization using DCT-based compression

To this end we have gathered enough mathematical bases and principles for (1D, 2D and 3D) data compression using DCT, so now it is time to formally establish the framework and operation procedures for topology optimization using DCT-based compression. Since 2D case can be degenerated by 3D case, in this section all formulations will be given in 3D space using 3D DCT and IDCT. The degeneration to 2D case is trivial so is omitted here but can be found in another paper by the authors [22].

##### 4.1. Traditional density method

Here we briefly review the process of solving topology optimization problems using traditional density method. As will be seen in Section 4.2, in DCDM the overall process is quite similar to that in traditional density method.

In traditional topology optimization method that is based on elemental fictitious density, the general mathematical formulation can be summarized as:

$$\begin{aligned} &\text{find} \quad \boldsymbol{\rho} \in (0, 1]^N \\ &\text{minimize} \quad f(\bar{\boldsymbol{\rho}}) \\ &\text{subject to} \quad h(\bar{\boldsymbol{\rho}}) \leq 0 \end{aligned} \quad (19)$$

where  $N = N_X N_Y N_Z$ ;  $N_X$ ,  $N_Y$  and  $N_Z$  are the number of elements in global  $X$ ,  $Y$  and  $Z$  directions, respectively;  $\boldsymbol{\rho}$  is the design variable while  $\bar{\boldsymbol{\rho}}$  is the elemental fictitious density and the pixel in spatial domain at the same time.

Usually in traditional density methods, the elemental fictitious density  $\bar{\boldsymbol{\rho}}$  is different from the design variable  $\boldsymbol{\rho}$ . Making  $\bar{\boldsymbol{\rho}}$  and  $\boldsymbol{\rho}$  the same would lead to some numerical instabilities such as the well-known checkerboard phenomenon [23,24].  $\bar{\boldsymbol{\rho}}$  is related to  $\boldsymbol{\rho}$  through density filter [19] and density projection [25].

In Eq. (19)  $f(\bar{\boldsymbol{\rho}})$  is the objective function while  $h(\bar{\boldsymbol{\rho}})$  is the constraint. Take the well-known static compliance minimization as an example, the objective is to minimize the static compliance under some given loads while the constraint is usually a restriction on material usage. So in this specific case the mathematical formulation can be given by

$$\begin{aligned} &\text{find} \quad \boldsymbol{\rho} \in (0, 1]^N \\ &\text{minimize} \quad \mathbf{Q}^{\top} \mathbf{F} \\ &\text{subject to} \quad \sum_{k=1}^N V_k \bar{\rho}_k \leq \gamma \sum_{k=1}^N V_k \end{aligned} \quad (20)$$



where  $\gamma$  is the material usage ratio and  $V_k$  is the volume of  $k$ th element; the global displacement vector  $\mathbf{Q} = [q_0, q_1, \dots, q_{n_{\text{dof}}-1}]^T \in \mathbb{R}^{n_{\text{dof}}}$  comes from the solution of static equilibrium equation:

$$\begin{aligned} \mathbf{K}\mathbf{Q} &= \mathbf{F}, \\ q_k &= \bar{q} \quad \text{for } k \in \Gamma_q. \end{aligned} \quad (21)$$

$\mathbf{K} \in \mathbb{R}^{n_{\text{dof}} \times n_{\text{dof}}}$  is the global stiffness matrix and  $\mathbf{F} \in \mathbb{R}^{n_{\text{dof}}}$  is the global load vector;  $n_{\text{dof}}$  is the number of dofs in the FEA model;  $\bar{q}$  is the given displacement (which is zero in this paper) while  $\Gamma_q$  is the given displacement boundary.  $\mathbf{K}$  is related to the elemental fictitious density  $\bar{\rho}$  through SIMP interpolation and finite elements assembly:

$$\mathbf{K} = \sum_{i=0}^{N-1} \bar{\rho}_i^3 \mathbf{K}_i^e, \quad (22)$$

where  $\mathbf{K}_i^e$  is the elemental stiffness matrix of  $i$ th element.

Sensitivity information  $\frac{\partial f}{\partial \rho}$  and  $\frac{\partial h}{\partial \rho}$  is needed for a gradient-based numerical optimization algorithm to find optimized solutions for Eq. (20):

$$\frac{\partial f}{\partial \rho_j} = \sum_{i=0}^{N-1} \frac{\partial f}{\partial \bar{\rho}_i} \frac{\partial \bar{\rho}_i}{\partial \rho_j}, \quad (23a)$$

$$\frac{\partial h}{\partial \rho_j} = \sum_{i=0}^{N-1} \frac{\partial h}{\partial \bar{\rho}_i} \frac{\partial \bar{\rho}_i}{\partial \rho_j}, \quad (23b)$$

where  $j = 0, 1, \dots, N-1$ ;  $\frac{\partial \bar{\rho}_i}{\partial \rho_j}$  depends on the choice of filter and projection while  $\frac{\partial f}{\partial \bar{\rho}_i}$  and  $\frac{\partial h}{\partial \bar{\rho}_i}$  can be given by

$$\frac{\partial f}{\partial \bar{\rho}_i} = \frac{\partial (\mathbf{Q}^T \mathbf{F})}{\partial \bar{\rho}_i} = -3\bar{\rho}_i^2 \mathbf{q}_i^T \mathbf{K}_i^e \mathbf{q}_i, \quad (24a)$$

$$\frac{\partial h}{\partial \bar{\rho}_i} = V_i \quad (24b)$$

where  $\mathbf{q}_i$  is the displacement components corresponding to  $i$ th element.

Here we comment that Eqs. (20) through (24) still hold true in the heat conduction problem where the objective is to minimize the heat resistance while the constraint is a restriction on material usage. In this case  $\mathbf{K}$  denotes the heat conduction matrix,  $\mathbf{F}$  denotes heat flux vector and  $\mathbf{Q}$  denotes the temperature of all nodes.

#### 4.2. DCT compression based density method (DCDM)

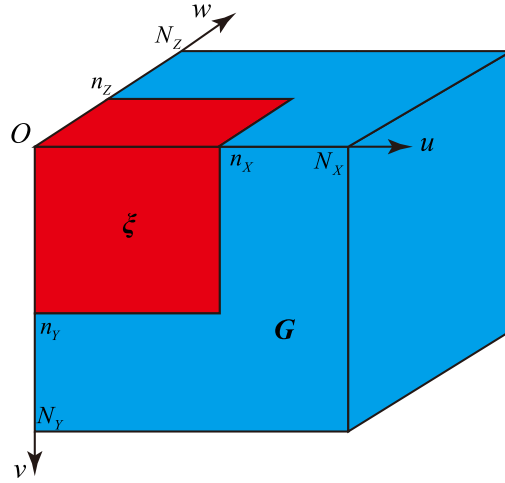
As remarked in the introduction section, the inefficiency of density-based methods partially comes from the inherent restriction that the number of design variables is precisely the same as the number of FEA meshes. By using DCDM, now it is possible to use only quite a few design variables:

$$\begin{aligned} &\text{find} \quad \boldsymbol{\xi} \in \mathbb{R}^n \\ &\text{minimize} \quad f(\bar{\rho}) \\ &\text{subject to} \quad h(\bar{\rho}) \leq 0 \end{aligned} \quad (25)$$

where  $\boldsymbol{\xi} = [\xi_0, \xi_1, \dots, \xi_{n-1}]^T$  is the design variable (cf. Eq. (26) and Fig. 5);  $n = n_X n_Y n_Z \ll N$  is the number of design variables;  $n_X \ll N_X$ ,  $n_Y \ll N_Y$  and  $n_Z \ll N_Z$  is the number of variables that will be used in global  $X$ ,  $Y$  and  $Z$  directions, respectively.

The map between  $\boldsymbol{\xi}$  and  $\rho$  then follows exactly the compression principles that are established in Section 2.2. The only difference lies in that the design variable  $\boldsymbol{\xi}$  belongs to the frequency domain and represents the nonzero low frequency components. The complete frequency domain matrix  $\mathbf{G}$  can be recovered by padding zeros (cf. Fig. 5):

$$\mathbf{G}(:, :, 0) = \left[ \begin{array}{ccc|c} \xi_0 & \cdots & \xi_{n_0-n_Y} & \mathbf{0} \\ \vdots & \ddots & \vdots & \\ \xi_{n_Y-1} & \cdots & \xi_{n_0-1} & \mathbf{0} \\ \hline & \mathbf{0} & & \mathbf{0} \end{array} \right], \quad (26a)$$



**Fig. 5.** The relation between design variables and DCT coefficients. The design variables  $\xi = [\xi_0, \xi_1, \dots, \xi_{n-1}]^T$  are the low frequency components of  $G$ . All the other high frequency components in  $G$  are set to zeros.

$$G(:, :, 1) = \left[ \begin{array}{ccc|c} \xi_{n_0} & \cdots & \xi_{2n_0-n_Y} & \mathbf{0} \\ \vdots & \ddots & \vdots & \\ \xi_{n_0+n_Y-1} & \cdots & \xi_{2n_0-1} & \mathbf{0} \\ \hline & \mathbf{0} & & \mathbf{0} \end{array} \right], \quad (26b)$$

$$\text{etc.}, \quad (26c)$$

which can be equivalently written in indicial notation (cf. Figs. 4 and 5):

$$G_{uvw} = \begin{cases} \xi_k, & \text{if } w = \left\lfloor \frac{k}{n_0} \right\rfloor, u = \text{mod}(k', n_Y), v = \frac{k' - u}{n_Y}; \\ 0, & \text{otherwise} \end{cases} \quad (27)$$

where  $u \in \{0, 1, \dots, N_X - 1\}$ ;  $v \in \{0, 1, \dots, N_Y - 1\}$ ;  $w \in \{0, 1, \dots, N_Z - 1\}$ ;  $k \in \{0, 1, \dots, n - 1\}$ ;  $n_0 = n_X n_Y$  is the number of design variables in each frontal slice (cf. Fig. 4);  $k' = k - wn_0$  is an auxiliary index;  $\text{mod}$  signifies the remainder after division;  $\lfloor \cdot \rfloor$  means rounding towards  $-\infty$ .

In short, Eq. (27) means the conversion between the indices and subscripts of a third tensor, which can simply be completed by using the *ind2sub* and *sub2ind* in MATLAB.

Then the pixels in spatial domain can be reconstructed by the 3D IDCT (cf. Eq. (16) and the Appendix section) of  $G^5$ :

$$\rho \equiv g = \text{IDCT}(G). \quad (28)$$

Since high frequency components are inherently filtered by the DCT-based compression, there is no need to use additional density filter or sensitivity filter, which can be seen as an advantage of the proposed method.

However, the density projection should still be used for two reasons. On one hand, the density projection manages to drive the elemental fictitious density to approach  $\{0, 1\}$  so that the final optimized topology is crisp. This motivation still holds in the proposed density method with DCT-based compression. On the other hand, due to the zero-padding procedure (and the bounds for the design variable  $\xi$ , as will be mentioned in Section 4.4), there is no guarantee that the pixels in spatial domain would still remain in  $(0, 1]$ . Therefore an additional density projection is in need to assure that the elemental fictitious densities still range in  $(0, 1]$ .

<sup>5</sup> Strictly speaking, the  $\rho$  in Eq. (28) is different from that in Eq. (19) because in Eq. (28)  $\rho$  is a third-order tensor while in Eq. (19)  $\rho$  is a vector, but by using column-wise concatenation one can freely transform between these two forms, so we do not explicitly emphasize this difference.

Here the (modified) smooth Heaviside projection [25] is used:

$$\bar{\rho}_{xyz} \equiv \bar{g}_{xyz} = \begin{cases} 0, & \text{if } g_{xyz} < 0; \\ 1, & \text{if } g_{xyz} > 1; \\ \frac{\tanh(\beta\eta) + \tanh[\beta(g_{xyz} - \eta)]}{\tanh(\beta\eta) + \tanh[\beta(1 - \eta)]}, & \text{otherwise} \end{cases} \quad (29)$$

where  $\eta = 0.5$  is the threshold for fictitious densities,  $\beta$  is a factor that can be gradually increased from 1 until 128 to drive  $\bar{g}_{xyz}$  to approach  $\{0, 1\}$  [25].

Now the reconstructed spatial pixels can be used in the famous SIMP interpolation scheme as in conventional density-based methods. The following steps are exactly the same as conventional density-based methods, so can be omitted here.

There exist some technical details that have not been addressed, i.e. the sensitivity analysis, the upper and lower bounds for the design variables  $\xi = [\xi_0, \xi_1, \dots, \xi_{n-1}]^T$  and the choice of initial solution. All these issues will be addressed in the next three subsections sequentially.

#### 4.3. Sensitivity information

Since all formulas are given in analytical expressions, the sensitivity information can simply be obtained by the chain rule:

$$\frac{\partial f}{\partial \xi_k} = \sum_{x,y,z} \frac{\partial f}{\partial \bar{g}_{xyz}} \frac{\partial \bar{g}_{xyz}}{\partial g_{xyz}} \frac{\partial g_{xyz}}{\partial \xi_k}, \quad (30)$$

where  $x \in \{0, 1, \dots, N_X - 1\}$ ,  $y \in \{0, 1, \dots, N_Y - 1\}$ ,  $z \in \{0, 1, \dots, N_Z - 1\}$ ;  $k \in \{0, 1, \dots, n - 1\}$ ;  $\frac{\partial f}{\partial \bar{g}_{xyz}}$  depends on the specific problem and is precisely the same as that in traditional density method (cf. Eq. (24));  $\frac{\partial \bar{g}_{xyz}}{\partial g_{xyz}}$  can be directly obtained from Eq. (29); the last term can be equivalently rewritten as (cf. Eq. (27) and Fig. 5):

$$\frac{\partial g_{xyz}}{\partial \xi_k} = \frac{\partial g_{xyz}}{\partial G_{uvw}}, \quad (31)$$

where  $w = \lfloor \frac{k}{n_0} \rfloor$ ,  $u = \text{mod}(k', n_Y)$ ,  $v = \frac{k' - u}{n_Y}$ ;  $n_0 = n_X n_Y$ ;  $k' = k - wn_0$ ; the ranges for  $x, y, z, k$  are exactly the same as that in Eq. (30).

From Eq. (16) it can be deduced that:

$$\frac{\partial g_{xyz}}{\partial G_{uvw}} = D_{ux}^{N_X} D_{vy}^{N_Y} D_{wz}^{N_Z}, \quad (32)$$

where  $x \in \{0, 1, \dots, N_X - 1\}$ ,  $y \in \{0, 1, \dots, N_Y - 1\}$ ,  $z \in \{0, 1, \dots, N_Z - 1\}$ ;  $u, v, w$  are the constants determined by  $k$  as mentioned above.

Eq. (32) warrants special attention since it reveals that  $\frac{\partial g_{xyz}}{\partial G_{uvw}}$  is a constant so as to be independent of the design variable  $\xi$  (and  $G$  as well). This observation is important to improve the efficiency for the sensitivity analysis: in each iteration only  $\frac{\partial f}{\partial \bar{g}_{xyz}}$  and  $\frac{\partial \bar{g}_{xyz}}{\partial g_{xyz}}$  need to be computed,  $\frac{\partial g_{xyz}}{\partial G_{uvw}}$  can be computed ahead of time only once. In this sense it can be concluded that the computational burden of the sensitivity analysis in each iteration is the same as that in traditional method. Moreover, the efficiency can be further improved if one notices that Eq. (32) can be written in matrix form, so the loop over the indices  $x, y, z$  can be avoided.

#### 4.4. Bounds for design variables

Since now we are optimizing the nonzero frequency components of the pixels, the accurate range for  $\xi$ , instead of simply using  $\mathbb{R}^n$  as shown in Eq. (19), should be determined so that the optimizer can find the optimal solution in a much more efficient manner.

$\xi$  is always equal to (the nonzero low frequency components of)  $G$ , so we only need to find the bounds for  $G_{uvw}$ , which can be interpreted as the following problem.

**Problem:** Consider the function DCT:  $(0, 1]^{N_X \times N_Y \times N_Z} \rightarrow \mathbb{R}$  given by

$$[g_{xyz}] \mapsto G_{uvw} = \sum_{x,y,z} D_{ux}^{N_X} D_{vy}^{N_Y} D_{wz}^{N_Z} g_{xyz}, \quad (33)$$

find the range for  $G_{uvw}$ .

**Solution:** Fortunately this problem belongs to the fundamental algebra, which gives the solution as:

$$\text{Range}(G_{uvw}) = \left[ \sum_{x,y,z} \min\{0, D_{ux}^{N_X} D_{vy}^{N_Y} D_{wz}^{N_Z}\}, \sum_{x,y,z} \max\{0, D_{ux}^{N_X} D_{vy}^{N_Y} D_{wz}^{N_Z}\} \right], \quad (34)$$

where  $u \in \{0, 1, \dots, n_X - 1\}$ ,  $v \in \{0, 1, \dots, n_Y - 1\}$ ,  $w \in \{0, 1, \dots, n_Z - 1\}$  while  $x \in \{0, 1, \dots, N_X - 1\}$ ,  $y \in \{0, 1, \dots, N_Y - 1\}$ ,  $z \in \{0, 1, \dots, N_Z - 1\}$ .

A special case is the bounds for  $G_{000} \equiv \xi_0$ , i.e. the DC component of the 3D image. Since when  $u = v = w = 0$ ,  $D_{ux}^{N_X} \equiv \frac{1}{\sqrt{N_X}}$ ,  $D_{vy}^{N_Y} \equiv \frac{1}{\sqrt{N_Y}}$ ,  $D_{wz}^{N_Z} \equiv \frac{1}{\sqrt{N_Z}}$ , the following concise result can be derived:

$$\text{Range}(G_{000}) = \left[ 0, \sqrt{N_X N_Y N_Z} \right]. \quad (35)$$

#### 4.5. Choice of initial solution

Almost all numerical optimization algorithms require the user to input an initial solution from which the optimization algorithms begin to search for better solutions. Generally, there are two choices of initial solution in topology optimization, i.e. random initial solution or uniform initial solution.

For random initial solution, one only needs to randomly select initial values for each  $\xi_k$  among the range given by Eq. (34). For uniform solution, i.e. the solution corresponding to

$$\text{vec}(\mathbf{g}) = \gamma[1, 1, \dots, 1]^T, \quad (36)$$

one can either substitute Eq. (36) into Eq. (33) to find the corresponding  $\xi$  or equivalently but in an easier manner, directly write down the following result based on physical intuition<sup>6</sup>:

$$\xi_k^0 = \begin{cases} \gamma \sqrt{N_X N_Y N_Z}, & \text{if } k = 0 \\ 0, & \text{otherwise,} \end{cases} \quad (37)$$

where  $k \in \{0, 1, \dots, n-1\}$ ;  $\gamma$  is the volume fraction of material that will be used in final design;  $\xi^0 = [\xi_0^0, \dots, \xi_{n-1}^0]^T$  is the initial trial solution for  $\xi$ .

## 5. Numerical examples

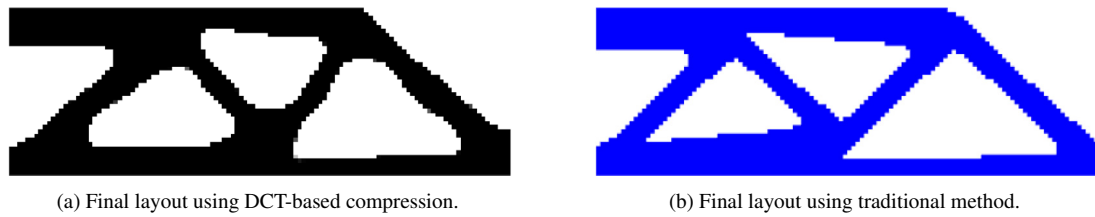
In this section the effectiveness and efficiency of the proposed method is demonstrated through several examples in both 2D and 3D space covering mechanical as well as heat conduction problems.

All the numerical results are computed using MATLAB 2017a on a Dell Precision T5610 workstation with two Intel(R) Xeon(R) CPU E5-2609 v2 processors. No explicit parallelization is used in all examples. The *e04wd* function in NAG library [26], which we find quite efficient and robust, is selected as the optimizer. The uniform initial solution (cf. Eq. (37)) is used throughout this section.

### 5.1. Minimization of static compliance: 2D cases

First some classical 2D examples that are extensively used as benchmarks in topology optimization research area are tested.

<sup>6</sup> For an image with uniform pixels, in frequency domain only the DC component  $G_{000}$  (cf. Eq. (35)) is nonzero while all others are zero.



**Fig. 6.** The optimized topologies for MBB beam.  $N_X = 120$ ,  $N_Y = 40$ ;  $n_X = 12$ ,  $n_Y = 5$ .

**Table 1**

Summary of the tested numerical examples.

Case	No. variables	obj. func.	Time (s)
MBB-DCT	60	$4.89 \times 10^5$	48.5
MBB	4 800	$4.68 \times 10^5$	403.7
cantilever-DCT	120	25.7311	60.3
cantilever	4 800	25.4537	316.4
box 3D-DCT	500	85.5481	2978.0
box 3D	32 000	84.3774	15 106.0
cantilever 3D-DCT	2 000	29.157	4543.0
cantilever 3D	31 250	22.290	11 356.0
heat-DCT	225	$1.42 \times 10^4$	36.5
heat	10 000	$1.18 \times 10^4$	1084.3
heat2-DCT	225	$2.26 \times 10^4$	27.64
heat2	10 000	$2.15 \times 10^4$	394.6353

### 5.1.1. MBB beam

The beam is symmetric in  $X$  axis and simply supported at its two ends. Due to the symmetry, only half structure is considered. Dimensions are: half span 120, height 40. Material parameters are:  $E = 1$ ,  $\nu = 0.3$ . Thickness  $t = 1$ . Half load  $0.5P = 50$  is applied at the center of top edge. Material usage  $\gamma = 0.5$ . The objective is to minimize the static compliance. When using traditional density method, the density filter is used, filter radius  $r_{\min} = 4$ . Final topologies are presented in Fig. 6 while detailed numerical information is listed in Table 1.

Due to the application of density projection (Eq. (29)), crisp final optimized layout can be obtained by both the traditional density method and the newly proposed method. Also, the final objective function values are quite close ( $4.68 \times 10^6$  vs.  $4.89 \times 10^6$ ) but the newly proposed method is much more efficient as shown in Table 1. It is interesting to notice that the final topology given by DCDM (DCT Compression based Density Method) presents no sharp corners, which would be beneficial when considering the stress of the structure. This observation is reasonable since sharp corners correspond to extremely high frequency components so is inherently filtered by the DCT-based compression.

### 5.1.2. Short cantilever

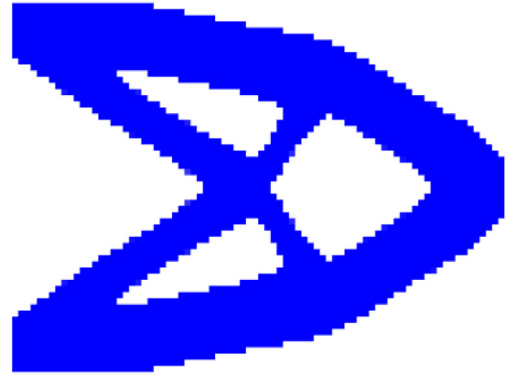
The beam is fixed at the left side while subjected to point-load in the middle of right side. Dimensions are: width 80, height 60. Material parameters are:  $E = 1$ ,  $\nu = 0.3$ . Thickness  $t = 1$ . Load  $P = -1$ . Material usage  $\gamma = 0.5$ . The objective is to minimize the static compliance. Final topologies are presented in Fig. 7 while detailed numerical information is listed in Table 1. In Fig. 7(b), the density filter corresponding to  $r_{\min} = 1.5$  is used. All the observations we made in the previous example still hold here.

## 5.2. Minimization of static compliance: 3D cases

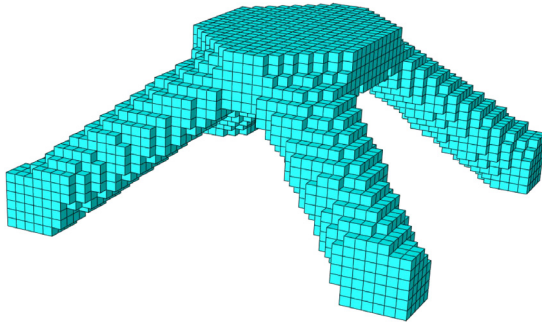
The examples presented in this subsection demonstrate the ability of the proposed method to deal with general 3D problems. We mention that since our codes are written in MATLAB, it is time-consuming to test extremely



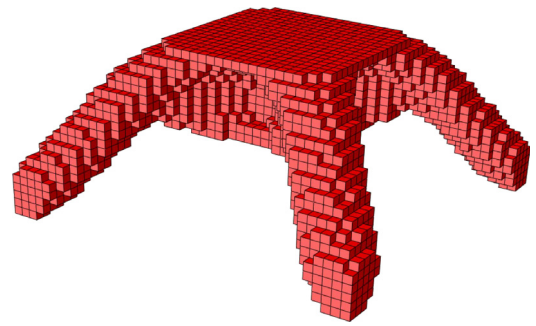
(a) Final layout using DCT-based compression.



(b) Final layout using traditional method.

**Fig. 7.** The optimized topologies for short cantilever.  $N_X = 80$ ,  $N_Y = 60$ ;  $n_X = 12$ ,  $n_Y = 10$ .

(a) Final layout using DCT-based compression.



(b) Final layout using traditional method.

**Fig. 8.** Final topologies of the cubic structure.  $N_X = N_Y = 40$ ,  $N_Z = 20$ ;  $n_X = n_Y = 10$ ,  $n_Z = 5$ .

large-scale 3D problems due to the FEA process.<sup>7</sup> However, the DCT-based compression is mesh-independent, just like the mesh-independent filter. As the mesh become finer, the compression ratio would increase as well. So in extremely large-scale problems the superiority of the proposed DCT-based compression would become more evident.

### 5.2.1. Box-like structure

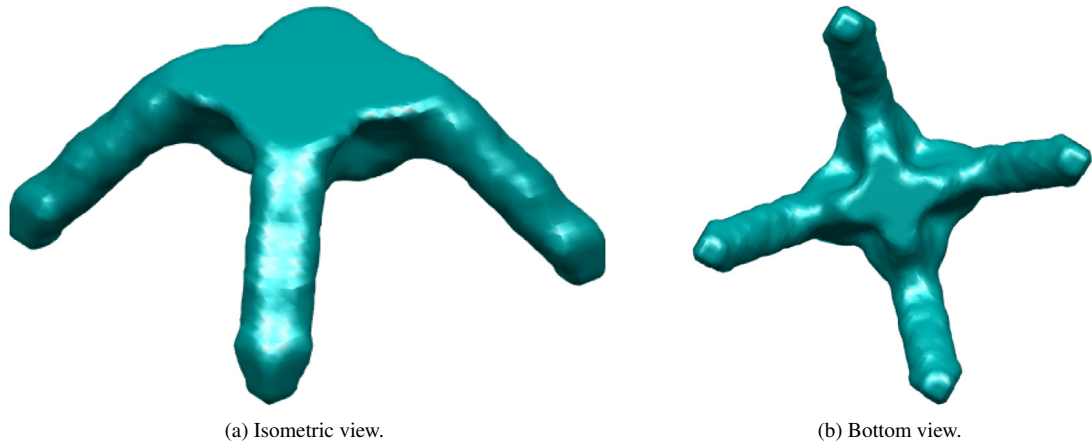
In this example we consider the static compliance of a box-like structure. Dimensions are: length 1000, width 1000, height 500. Material properties are:  $E = 200 \times 10^3$ ,  $\nu = 0.3$ . Material usage  $\gamma = 15\%$ . A concentrated load  $1 \times 10^4$  is applied at the center of top surface. The box is simply supported at the four corners in the bottom surface. When using traditional density method, the density filter corresponding to  $r_{\min} = 37.5$  is applied. Final layouts obtained by DCDM and the traditional density method are shown in Fig. 8 while detailed numerical information can be found in Table 1.

As can be seen, the two topologies shown in Fig. 8 are quite similar though some minor differences exist. The reduction in the number of design variables (32 000 vs. 500) and computation time (2978 s vs. 15 106 s) is quite considerable. The postprocessed topology for Fig. 8(a) is shown in Fig. 9.

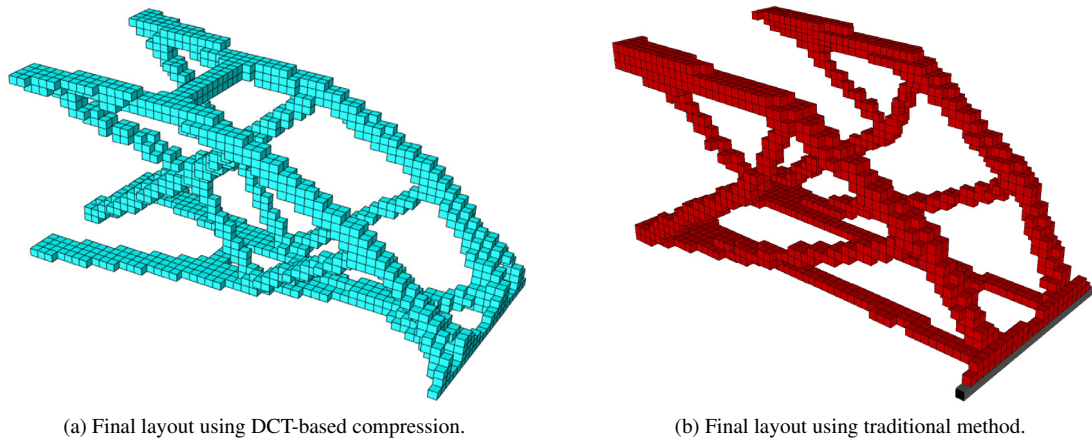
### 5.2.2. 3D cantilever

In this example we consider the compliance minimization of a cube. Dimensions are: length 2000, width 1000, height 1000. Material properties are:  $E = 200 \times 10^3$ ,  $\nu = 0.3$ . Material usage  $\gamma = 5\%$ . Uniform load 2500 N is

<sup>7</sup> Note that in the proposed method, the FEA process follows exactly the way in traditional method with same computation cost.



**Fig. 9.** Final layout of the box-like structure after postprocessing.



**Fig. 10.** Final layout for 3D cantilever.  $N_X = 50$ ,  $N_Y = N_Z = 25$ ;  $n_X = 20$ ,  $n_Y = n_Z = 10$ .

applied at the bottom right edge. This example was used in [27] to demonstrate the power of parallelization computing, in this paper no explicit parallelization is used so a much coarse mesh will be adopted. When using traditional density method, the density filter corresponding to  $r_{\min} = 60$  is applied. Final layouts obtained by the traditional density method and DCDM are shown in Fig. 10 while detailed numerical information is listed in Table 1. In Fig. 10 one can find that only minor differences occur between the two final designs. The postprocessed topology<sup>8</sup> for Fig. 10(a) is shown in Fig. 11.

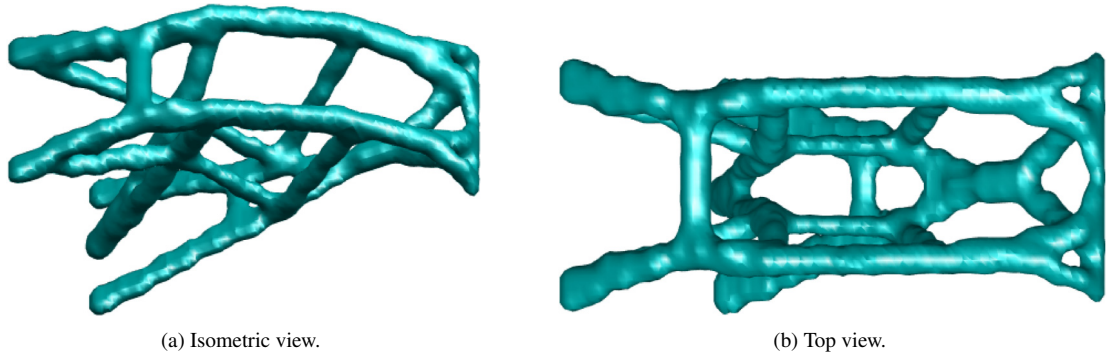
Notice that in this example the layout of the final optimized design is a little bit complicated, so the compression ratio (with respect to both computation time and number of design variables) is not as large as that in previous examples.

### 5.3. Minimization of the heat resistance of a square plate

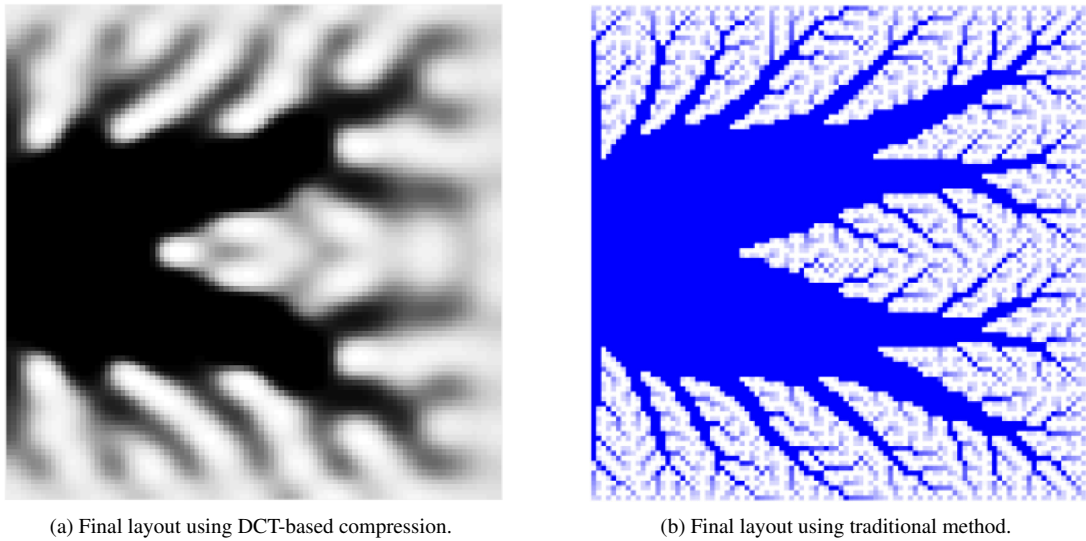
In this example we consider the heat conduction of a square plate. Length of each side is 100. Heat sink is located at the middle of left side. Uniform internal heat source  $Q = 0.01$  is applied on the whole plate (except the heat sink).

<sup>8</sup> The postprocessing techniques include the smoothing of density field and computing the isosurface of smoothed density field to obtain clear and crisp topology boundary.





**Fig. 11.** Final layout for 3D cantilever after postprocessing.



**Fig. 12.** The optimized topologies for heat conduction problem.  $N_X = N_Y = 100$ ,  $n_X = n_Y = 15$ .

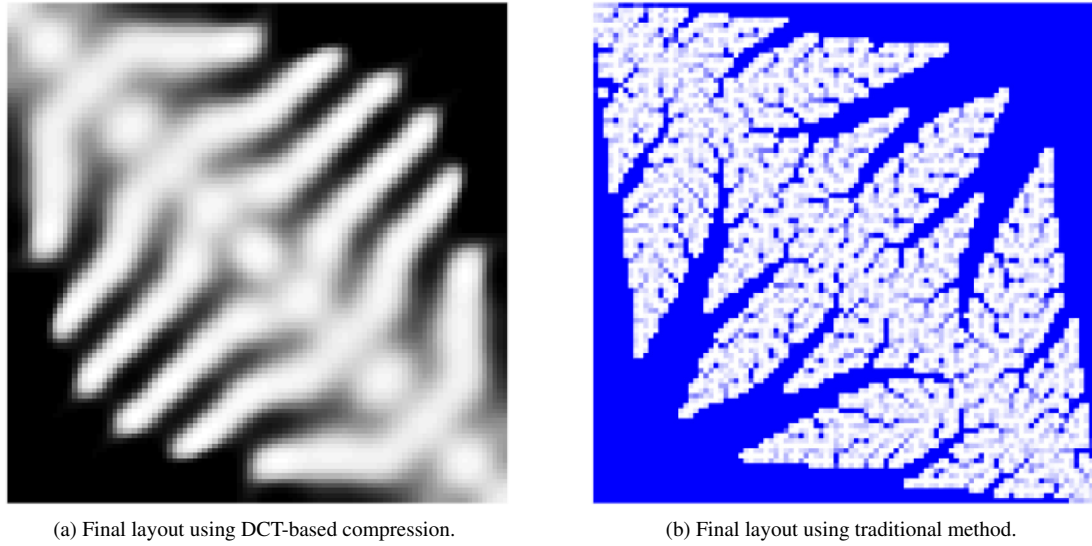
Thermal conductivity is  $\kappa = 1$ . Material usage  $\gamma = 0.5$ . The objective is to minimize the heat resistance. When using traditional density method, the density filter corresponding to  $r_{\min} = 2.1$  is applied. Final topologies are presented in Fig. 12 while detailed numerical information is listed in Table 1. In Fig. 13 another example is shown where the heat sinks are located at the top right corner and the bottom left corner.

In Figs. 12 and 13, the major outlines of the two final designs are quite similar but the details (i.e. the tiny branches) present in Figs. 12(b) and 13(b) cannot be found in Figs. 12(a) and 13(a) since the tiny branches correspond to high frequency components as well. Recall that in engineering practice, the very slim branches can hardly be machined, so we think that the differences between these two results can be tolerated.

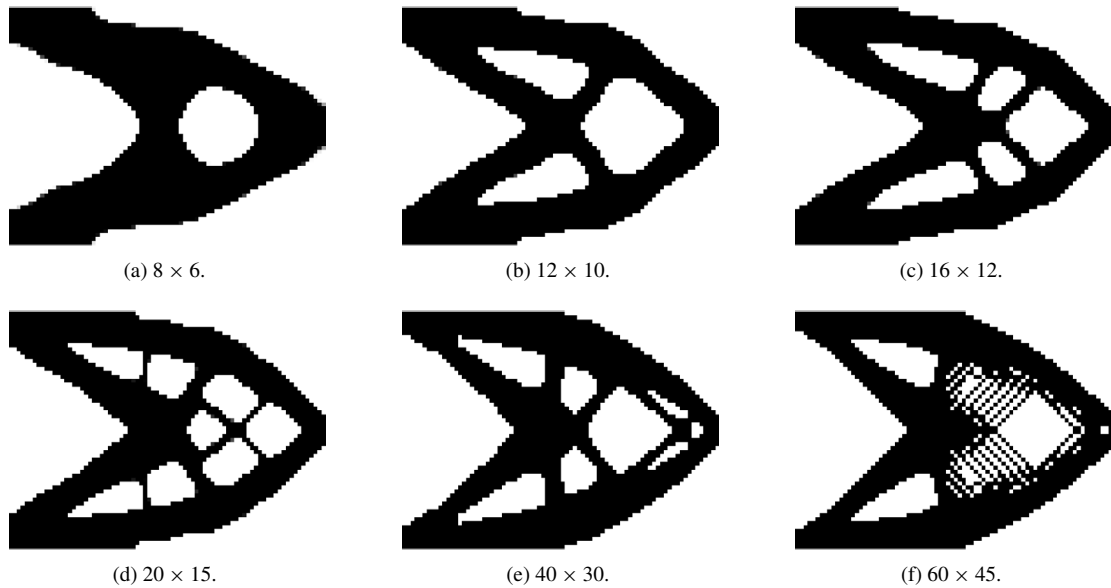
## 6. Discussions

### 6.1. On the choice of $n$

The number  $n = n_X n_Y n_Z$  determines how many (nonzero) frequency components will be retained in DCDM. If  $n = N = N_X N_Y N_Z$ , then no frequency component would be discarded, which implies that we were solving the topology optimization problems without any (density or sensitivity) filter. So it follows that  $n$  has some similar effects



**Fig. 13.** The optimized topologies for heat conduction problem.  $N_X = N_Y = 100$ ,  $n_X = n_Y = 15$ .



**Fig. 14.** The influence of  $n$  on final optimized design. The numbers in each label means  $n_X \times n_Y$ .  $N_X = 80$ ,  $N_Y = 60$ .

as the radius of filter. According to our experiences of using filters in density-based methods, it can be inferred that if  $n$  is too small, then only some main features can be captured by the topology optimization while the detailed features (that are also of interest) cannot be found. On the contrary, if  $n$  is too large then the checkerboard phenomenon which corresponds to the high frequency components would occur.

A simple numerical experiment is designed to verify the above inferences using the cantilever beam example. Material properties and boundary conditions are the same as that in Section 5.1.2. The final optimized designs corresponding to different choices of  $n$  are listed in Fig. 14 while detailed numerical information is collected in Table 2. Fig. 15 shows the relation between the number of design variables and the computation time.

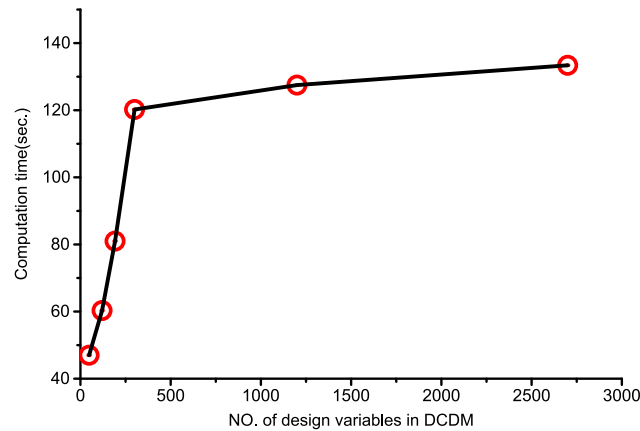


Fig. 15. Relation between NO. of design variables and computation time.

Based on Figs. 14, 15 and Table 2, some conclusions can be made on the influence of  $n$ :

- (1) The number of design variables (i.e. the number of nonzero frequency components) has similar effects as the filter radius in conventional density method. If  $n$  is small (cf. Fig. 14(a)), final design would be somewhat simple, i.e. some detailed geometrical information may be lost. On the contrary, if  $n$  is too large (cf. Fig. 14(f)), the checkerboard phenomenon would come into existence.
- (2) The number of design variables is positive correlative to the computation time. As  $n$  decreases, the computation time decreases.
- (3) The computation time is not linearly dependent on the number of design variables. This is reasonable since the time cost for FEA calculation is not reduced in the current version of DCDM. So if the compression ratio is not considerable, there would be no much differences in the computation time.
- (4) More design variables do not necessarily lead to better objective function values. Specifically, when using  $60 \times 45$  design variables, the final objective function value is the worst among all cases (though there are no big differences in all the tested cases). This may be due to the non-convexity of the optimization problem.
- (5) It is unpractical to expect too high compression ratio. The mesh  $12 \times 10$  corresponds to  $CR = 40$ , which can lead to reasonable final layout at reasonable cost. If the mesh  $8 \times 6$  is used, the number of design variables is cut down by 60% but the computation time is cut down by only 20%.

## 6.2. Price paid by using DCDM

Although the benefits of using DCDM in topology optimization problems seem to be exciting, researchers in this field should keep in mind that the DCT-based methods are all categorized as lossy compression methods in image compression. So by using the DCDM we do pay some price:

- (1) For problems having good properties such as convexity in spatial domain (though this is very unusual in fact), there is no guarantee that such good properties would also hold in frequency domain. So in general no global optimality can be assured by using the DCDM.
- (2) In DCDM the design variables take effect in frequency domain, so each design variable would have influence on *all* spatial pixels, which is totally different from the traditional density method where each design variable has influence on *one* pixel only. So it is somewhat difficult for the proposed method to precisely tune the intensity of some specific pixels. It follows that in general the final objective function values obtained by DCDM would be slightly higher than the ones by traditional methods.
- (3) Throughout this paper we take it as a premise that the tiny features (such as the very slim branches in Fig. 12) are of little use. In many occasions this assumption can be accepted as reasonable considering the ease for manufacturing, local stability, reliability, and so on. But if the tiny features are important, the possible gain

**Table 2**  
Influence of the number of design variables  $n$ .

$n_X \times n_Y$	No. variables	obj. func.	Time (s)
$60 \times 45$	2700	28.06	133.39
$40 \times 30$	1200	27.50	127.49
$20 \times 15$	300	25.48	120.22
$16 \times 12$	192	25.71	80.99
$12 \times 10$	120	25.73	60.3
$8 \times 6$	48	27.29	47.0

when using the DCDM would be quite limited since in that case  $n \approx N$ , i.e. there is no much redundancy for the data compression.

- (4) The fast algorithms for DCT and IDCT are only valid in equal-spaced meshes, so only rectangular (in 2D) and cubic (in 3D) domains can be dealt with so far. Therefore an improved version of DCDM, which is able to manage the more general case where irregular design domain may exist, is needed in the future.

## 7. Conclusions

In the field of topology optimization, although the density-based methods are regarded as the ones that are most popular, robust and easy to use until now, the efficiency of the methods is still a bottleneck for their applications in large scale engineering problems since a huge number of design variables need to be dealt with. In this paper we make some preliminary attempts to improve the efficiency of density-based methods by using DCT-based compression. The amount of design variables may be reduced to a great extent. Numerical results covering some typical problems in this community show that quite exciting benefits can be obtained by using the proposed DCDM: the efficiency is phenomenally improved; the additional filtering procedure can be circumvented while the checkerboard is avoided; sharp corners can be inherently eliminated without explicit control. Considering its good performance and easy implementation, it is hoped that the proposed DCDM method could be favored by the community of topology optimization.

## Acknowledgments

The research is supported by NSFC (11372154, 11772170) which is gratefully acknowledged by the authors. The authors would also like to thank Mr. Daniel White with whom the helpful discussions during WCSMO12 at Braunschweig, Germany inspires the first author to come up with the basic idea of this paper.

## Appendix

In this appendix we attempt to find the matrix operations for 3D DCT. The Einstein summation convention is adopted in the appendix.

As mentioned, the 3D DCT can be achieved by sequentially applying 1D DCT to each column, row and tube of the matrix  $[g_{xyz}]$ . Refer to Eq. (4), the 1D DCT can simply be seen as multiplying the transformation matrix from the left of  $[g_x]$  in spatial domain. Using the knowledge from tensor analysis, one knows that the matrix multiplication is equal to dot product between tensors. So when applying 1D DCT to the columns of  $[g_{xyz}]$ , we can simply multiply  $D^{N_X}$  to each column of  $[g_{xyz}]$ :

$$\mathbf{G}^X = D^{N_X} \mathbf{g} = D_{ux}^{N_X} g_{xyz} \mathbf{e}_u \mathbf{e}_y \mathbf{e}_z \quad (38)$$

where  $\mathbf{e}_i$  stands for the Cartesian basis.

Now we deal with the rows of  $\mathbf{G}^X$ , so the index  $y$  should be moved to the first place, which means the transpose between  $u$  and  $yz$  is needed:

$$\bar{\mathbf{G}}^X = (\mathbf{G}^X)^{\top_{[231]}} = D_{ux}^{N_X} g_{xyz} \mathbf{e}_y \mathbf{e}_z \mathbf{e}_u \quad (39)$$

where the notion  $\top_{[231]}$  denotes that original second, third and first index are rearranged as the first, second and third index, respectively.

Now the 1D DCT can be applied to the columns of  $\bar{\mathbf{G}}^X$ :

$$\mathbf{G}^{XY} = \mathbf{D}^{N_Y} \bar{\mathbf{G}}^X = \mathbf{D}_{vy}^{N_Y} \mathbf{D}_{ux}^{N_X} g_{xyz} \mathbf{e}_v \mathbf{e}_z \mathbf{e}_u. \quad (40)$$

Now transpose between  $z$  and  $vu$  to push forward the index  $z$ :

$$\bar{\mathbf{G}}^{XY} = (\mathbf{G}^{XY})^{\top[231]} = \mathbf{D}_{vy}^{N_Y} \mathbf{D}_{ux}^{N_X} g_{xyz} \mathbf{e}_z \mathbf{e}_u \mathbf{e}_v. \quad (41)$$

Continue to apply 1D DCT to the index  $z$ :

$$\mathbf{G}^{XYZ} = \mathbf{D}^{N_Z} \bar{\mathbf{G}}^{XY} = \mathbf{D}_{wz}^{N_Z} \mathbf{D}_{vy}^{N_Y} \mathbf{D}_{ux}^{N_X} g_{xyz} \mathbf{e}_w \mathbf{e}_u \mathbf{e}_v. \quad (42)$$

Finally notice that the indices are not arranged in the correct order  $uvw$ , so another transpose is needed:

$$\mathbf{G} = (\mathbf{G}^{XYZ})^{\top[231]} = \mathbf{D}_{wz}^{N_Z} \mathbf{D}_{vy}^{N_Y} \mathbf{D}_{ux}^{N_X} g_{xyz} \mathbf{e}_u \mathbf{e}_v \mathbf{e}_w. \quad (43)$$

Combining all the above equations leads to:

$$\mathbf{G} = \left( \mathbf{D}^{N_Z} \left( \mathbf{D}^{N_Y} \left( \mathbf{D}^{N_X} \mathbf{g} \right)^{\top[231]} \right)^{\top[231]} \right)^{\top[231]}. \quad (44)$$

The matrix operations for 3D IDCT can be found in precisely the same manner,

$$\mathbf{g} = \left( (\mathbf{D}^{N_Z})^{\top} \left( (\mathbf{D}^{N_Y})^{\top} \left( (\mathbf{D}^{N_X})^{\top} \mathbf{G} \right)^{\top[231]} \right)^{\top[231]} \right)^{\top[231]} \quad (45)$$

and the derivation process is omitted here.

To close this appendix, we mention that in MATLAB all the above matrix operations can be circumvented. Only one piece of code is needed to complete the 3D DCT or 3D IDCT:

$$\mathbf{G} = \text{dct}(\text{dct}(\text{dct}(\mathbf{g}, [], 1), [], 2), [], 3); \quad (46)$$

$$\mathbf{g} = \text{idct}(\text{idct}(\text{idct}(\mathbf{G}, [], 1), [], 2), [], 3); \quad (47)$$

## References

- [1] M. Bendsoe, N. Kikuchi, Generating optimal topologies in structural design using a homogenization method, *Comput. Methods Appl. Mech. Engrg.* 71 (1988) 197–224.
- [2] M. Bendsoe, Optimal shape design as a material distribution problem, *Struct. Optim.* 1 (1989) 193–202.
- [3] G. Rozvany, M. Zhou, T. Birker, Generalized shape optimization without homogenization, *Struct. Optim.* 4 (1992) 250–254.
- [4] O. Sigmund, A 99 line topology optimization code written in matlab, *Struct. Multidiscip. Optim.* (2001).
- [5] X. Huang, M. Xie, *Evolutionary Topology Optimization of Continuum Structures: Methods and Applications*, John Wiley & Sons, 2010.
- [6] Y.M. Xie, G. Steven, A simple evolutionary procedure for structural optimization, *Comput. & Structures* 49 (5) (1993) 885–896.
- [7] M.Y. Wang, X. Wang, D. Guo, A level set method for structural topology optimization, *Comput. Methods Appl. Mech. Engrg.* 192 (1) (2003) 227–246. [http://dx.doi.org/10.1016/S0045-7825\(02\)00559-5](http://dx.doi.org/10.1016/S0045-7825(02)00559-5).
- [8] X. Guo, W. Zhang, W. Zhong, Doing topology optimization explicitly and geometrically: A new moving morphable components based framework, *J. Appl. Mech.* (2014).
- [9] R.T. Haftka, Z. Gürdal, *Elements of Structural Optimization*, Vol. 11, Springer Science & Business Media, 2012.
- [10] J. Park, A. Sutradhar, A multi-resolution method for 3d multi-material topology optimization, *Comput. Methods Appl. Mech. Engrg.* 285 (2015) 571–586. <http://dx.doi.org/10.1016/j.cma.2014.10.011>.
- [11] W. Zhang, J. Yuan, J. Zhang, X. Guo, A new topology optimization approach based on moving morphable components (mmc) and the ersatz material model, *Struct. Multidiscip. Optim.* 53 (6) (2016) 1243–1260.
- [12] X. Guo, W. Zhang, J. Zhang, J. Yuan, Explicit structural topology optimization based on moving morphable components (mmc) with curved skeletons, *Comput. Methods Appl. Mech. Engrg.* 310 (2016) 711–748. <http://dx.doi.org/10.1016/j.cma.2016.07.018>.
- [13] W. Zhang, D. Li, J. Zhang, X. Guo, Minimum length scale control in structural topology optimization based on the moving morphable components (mmc) approach, *Comput. Methods Appl. Mech. Engrg.* 311 (2016) 327–355. <http://dx.doi.org/10.1016/j.cma.2016.08.022>.
- [14] R.G. Gonzalez, R.E. Woods, *Digital Image Processing*, Publishing House of Electronics Industry, 2007.
- [15] K.R. Rao, P. Yip, *Discrete Cosine Transform: Algorithms, Advantages and Applications*, Academic Press, 1990.
- [16] W. Zhang, J. Chen, X. Zhu, J. Zhou, D. Xue, X. Lei, X. Guo, Explicit three dimensional topology optimization via moving morphable void (mmv) approach, *Comput. Methods Appl. Mech. Engrg.* 322 (2017) 590–614. <http://dx.doi.org/10.1016/j.cma.2017.05.002>.
- [17] W. Zhang, J. Song, J. Zhou, Z. Du, Y. Zhu, Z. Sun, X. Guo, Topology optimization with multiple materials via moving morphable component (mmc) method, *Internat. J. Numer. Methods Engrg.* (2017). <http://dx.doi.org/10.1002/nme.5714>.

- [18] D. Gottlieb, C.-W. Shu, On the gibbs phenomenon and its resolution, *SIAM Rev.* 39 (4) (1997) 644–668.
- [19] B. Bourdin, Filters in topology optimization, *Internat. J. Numer. Methods Engrg.* 50 (9) (2001) 2143–2158.
- [20] O. Sigmund, Morphology-based black and white filters for topology optimization, *Struct. Multidiscip. Optim.* 33 (4–5) (2007) 401–424. <http://dx.doi.org/10.1007/s00158-006-0087-x>.
- [21] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, *SIAM Rev.* 51 (3) (2009) 455–500. <http://dx.doi.org/10.1137/07070111X>.
- [22] P. Zhou, J. Du, Z. Lü, Highly efficient density-based topology optimization using dct-based digital image compression, *Struct. Multidiscip. Optim.* 57 (1) (2018) 463–467. <http://dx.doi.org/10.1007/s00158-017-1840-z>.
- [23] M. Zhou, Y. Shyy, H. Thomas, Checkerboard and minimum member size control in topology optimization, *Struct. Multidiscip. Optim.* 21 (2) (2001) 152–158.
- [24] O. Sigmund, J. Petersson, Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima, *Struct. Optim.* 16 (1) (1998) 68–75.
- [25] F. Wang, B.S. Lazarov, O. Sigmund, On projection methods, convergence and robust formulations in topology optimization, *Struct. Multidiscip. Optim.* 43 (6) (2011) 767–784. <http://dx.doi.org/10.1007/s00158-010-0602-y>.
- [26] The Numerical Algorithms Group, The nag library, 2017. URL <https://www.nag.com>.
- [27] N. Aage, B.S. Lazarov, Parallel framework for topology optimization using the method of moving asymptotes, *Struct. Multidiscip. Optim.* 47 (4) (2013) 493–505. <http://dx.doi.org/10.1007/s00158-012-0869-2>.