To appear in *Computer Methods in Biomechanics and Biomedical Engineering*
Vol. 00, No. 00, Month 20XX, 1–14

A Data-Driven Investigation and Estimation of Optimal Topologies Under Variable Loading Configurations

Erva Ulu, Rusheng Zhang, Levent Burak Kara*

Carnegie Mellon University

(*Received 00 Month 20XX; accepted 00 Month 20XX*)

Topology optimization problems involving structural mechanics are highly dependent on the design constraints and boundary conditions. Thus, even small alterations in such parameters require a new application of the optimization routine. To address this problem, we examine the use of known solutions for predicting optimal topologies under a new set of design constraints. In this context, we explore the feasibility and performance of a data-driven approach to structural topology optimization problems. Our approach takes as input a set of images representing optimal 2-D topologies, each resulting from a random loading configuration applied to a common boundary support condition. These images represented in a high dimensional feature space are projected into a lower dimensional space using component analysis. Using the resulting components, a mapping between the loading configurations and the optimal topologies is learned. From this mapping, we estimate the optimal topologies for novel loading configurations. The results indicate that when there is an underlying structure in the set of existing solutions, the proposed method can successfully predict the optimal topologies in novel loading configurations. In addition, the topologies predicted by the proposed method can be used as effective initial conditions for conventional topology optimization routines, resulting in substantial performance gains. We discuss the advantages and limitations of the presented approach and show its performance on a number of examples.

Keywords: data-driven design; topology optimization; dimensionality reduction

1. Introduction

Efficient use of material is a key priority for designers in many industries including automotive, aerospace and consumer product industries (Bendsoe and Sigmund 2004; Schramm and Zhou 2006; Richardson, Coelho and Adriaenssens 2010). Optimizing material layout to satisfy a specific performance criteria, i.e. topology optimization is thus a crucial part of engineering design process. With recent advances in manufacturing technologies, topology optimization now attracts even more attention (Schramm and Zhou 2006). So far, various optimization algorithms including genetic algorithms, method of moving asymptotes, level sets and topological derivatives have been studied for structural topology optimization (Bendsoe and Sigmund 2004; Chapman, Saitou and Jakiela 1994; Jakiela, Chapman, et al. 2000; Aage and Lazarov 2013; Dijk, Maute, et al. 2013; Norato, Bendsoe, et al. 2007).

Although structural optimization algorithms are becoming computationally more efficient with time, the need for a large number of iterations can rarely be avoided due to the essence of optimization theory. Even with very small alterations in the design constraints, a structurally optimum topology can not be predicted directly by a human from physical principles due to the complex nature of the problem. Based on this observation, we explore how known solutions to topology optimization problems can be exploited to generate a new design for a novel set of loading con-

*Corresponding author. Email: lkara@cmu.edu

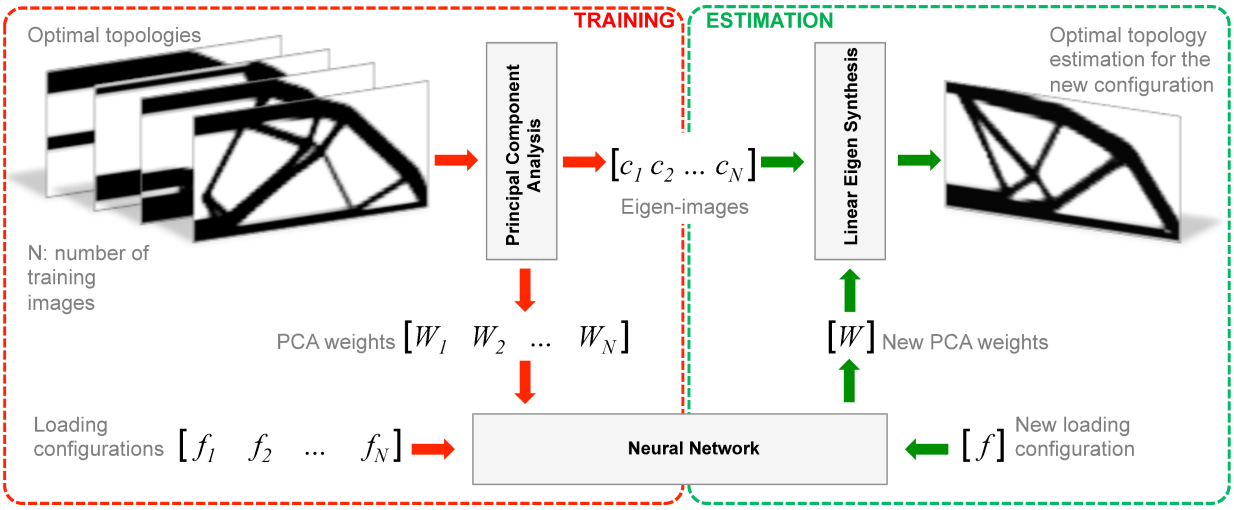

Figure 1.: Overview of our approach for optimal topology estimation.

figurations. Here, the main challenge is to find a mapping between design constraints and the resulting optimal topologies. With this motivation, we present a data-driven approach to topology optimization involving structural mechanics and explore its feasibility and performance.

Our approach takes as input a set of images representing optimal 2D topologies, each resulting from a conventional optimization method, and generates an optimal topology estimation for a novel set of design constraints (Fig. 1). In this study, only the variation in loading configurations is explored under a fixed set of structural boundary conditions. However, the application of the presented method is not limited to this choice, because design constraints can be expanded to accommodate any changes in the boundary conditions as long as the size of the overall design domain is kept the same. In the proposed method, the set of input images (known optimal topologies) which are represented in a high dimensional image space are projected onto a lower dimensional space using Principal Component Analysis (PCA). Once the dimensionality is reduced, a mapping between the loading configurations and the optimal topologies represented as PCA component weights is computed using a feed-forward neural network. Using the trained mapping, we estimate the PCA component weights for a novel loading configuration, and use the resulting estimation to synthesize a solution in the image space. This image represents our estimation of the optimal topology, given a novel loading configuration.

The primary goal of this study is to explore the feasibility and effectiveness of a data-driven approach to structural topology optimization problems. Our results show that the proposed method can successfully predict the optimal topologies in different problem settings, but the results are sensitive to the complexity and the size of the design space dictated by the loading configurations. However, independent of the problem complexity, a practical advantage of the proposed system is that the resulting topology estimations serve as effective initial conditions that facilitate faster convergence in conventional topology optimization problems.

2. Literature Review

In this section, we review the relevant literature on structural topology optimization techniques, use of data analysis and dimensionality reduction approaches as well as data mapping methods.

Structural Topology Optimization: Topology optimization is one of the most powerful technologies in structural design (Bendsoe and Sigmund 2004; Dijk, Maute, et al. 2013). It optimizes the shape and material connectivity of a domain through the use of finite element methods together with various optimization techniques (Norato, Bendsoe, et al. 2007).

Density-based topology optimization approaches including homogenization methods (Bendsoe

and Kikuchi 1988; Suzuki and Kikuchi 1991) and solid isotropic microstructure with penalty (SIMP) methods (Bendsoe 1989; Rozvany, Zhou and Birker 1992) are one of the most popular methods in the literature. These methods approach topology optimization in a way that defines geometry by optimizing material distribution in the domain. A detailed review on density based topology optimization methods can be found in (Hassani and Hinton 1998; Rozvany 2001, 2009). Another approach for structural topology optimization is based on topological derivatives and level-sets (Norato, Bendsoe, et al. 2007; Sethian and Wiegmann 2000; Wang, Wang and Guo 2003). The optimization process utilizes the implicit description of the boundary to numerically represent the geometry. A recent work (Dijk, Maute, et al. 2013) discusses the level-set based topology optimization methods more deeply. In (Rozvany 2009), topological derivative and level-set based methods in the literature are claimed to be very promising although they are not widely embraced by industries. Aside from the above methods, evolutionary approaches are also used for topology optimization, e.g. (Chapman, Saitou and Jakiela 1994; Jakiela, Chapman, et al. 2000). However, the use of genetic algorithms are computationally expensive, thus they are suitable for only small scale problems (Rozvany 2009).

Since topology optimization is an iterative and computationally demanding process, an efficient implementation of the above mentioned methods in various programming languages is also important for designers. In (Andreassen, Clausen, et al. 2011; Sigmund 2001), authors present two different versions of an efficient MATLAB code for structural topology optimization of classical Messerschmitt-Blkow-Blohm (MBB) beam problem. As an optimization technique, they implemented an available SIMP approach with slight modifications involving filters. In our approach, we utilize the available code in (Andreassen, Clausen, et al. 2011) to generate the initial optimized topologies for different loading conditions as a way to generate the pool of training data.

Data Analysis and Dimensionality Reduction: In data-driven methods, a pre-analysis of available data to extract informative characteristics is essential, especially for large multivariate data sets. Such methods are commonly used in engineering design and computer science, e.g. (Sirovich and Kirby 1987; Kirby and Sirovich 1990; Turk and Pentland 1991; Allen, Curless and Popović 2003; Yumer and Kara 2011, 2012). Although the design approach is not data driven, dimensionality reduction idea is also used in structural topology optimization in (Guest and Smith Genut 2010) to reduce the computational cost by decreasing the number of independent design variables.

In data driven design context, the most commonly used dimensionality reduction methods include principal component analysis (PCA) (Jolliffe 2005), multidimensional scaling (MDS) (Cox and Cox 1994), Isomaps (Tenenbaum 1998) and locally linear embedding (LLE) (Roweis and Saul 2000). PCA is an eigenvector based approach that uses an orthogonal transformation to convert the original data into linearly independent components. Dimensionality reduction is accomplished by representing data in terms of the linearly independent components that best explain the variance in the data. In MDS, high dimensional data is embedded into low dimensional space in such a way that pairwise distances between data points are preserved. Isomaps aim to preserve the geodesic distances in the manifold formed by the data. LLE is a neighborhood-preserving dimensionality reduction method. It projects high-dimensional data into lower dimensional global coordinates by utilizing different linear embeddings for each data point locally. In the proposed work, we use PCA to analyze the dominant characteristics of our data set and to reduce the dimensionality accordingly. However, the aforementioned dimensionality reduction methods could be adopted into the workflow of the proposed techniques without loss of generality.

One important aspect of the proposed work is the mapping between an input configuration (in our case the loading configuration) and the resulting optimal topology. Note that a PCA-based learning and topology reconstruction is readily implementable with the available training images. However, the key need is to be able to specify a novel loading configuration, from which the optimal topology can be estimated. In previous work, most methods employ a linear mapping between the input feature vectors and the resulting PCA reconstructions (Allen, Curless and Popović 2003; Blanz and Vetter 1999). However, the relationship between the input loading configurations and

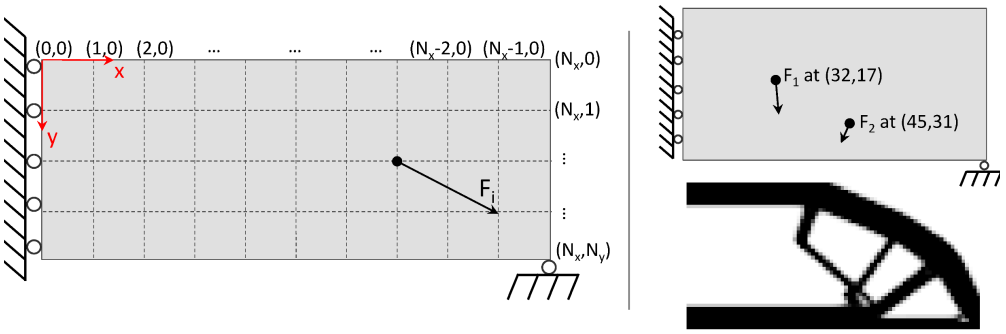

Figure 2.: Left: N_x-by-N_y design domain for topology optimization problem. Right: Example loading configuration and resulting optimal topology. Multiple external forces, F_i, can be applied to the beam at the nodes represented by (x_i, y_i) coordinates.

the resulting topology reconstructions in our domain is highly non-linear as demonstrated in the following sections. To address this challenge, we present a mapping technique that uses feed-forward neural networks. This generative method provides a significant improvement over linear regression models by covering non-linearities automatically without requiring any explicit information about the design space complexity.

3. Problem Formulation

We illustrate our approach using the Messerschmitt-Blkow-Blohm (MBB) beam problem, a classical problem in topology optimization. The rectangular beam is represented by an N_x-by-N_y image as illustrated in Fig. 2. The design domain is discretized by square finite elements each of which corresponds to a pixel in the gray-scale images. A number of external forces, F_i $(i = 1, ..., k)$, can be applied to the beam at the nodes represented by (x_i, y_i) coordinates. Applied forces can be in any direction, i.e. they can have both horizontal and vertical components with magnitudes ranging between $[0, 1]$. Boundary support conditions are shown in Fig. 2. Note that this model is used to facilitate discussions; the following sections will demonstrate results on variations of the domain, boundary conditions and loading configurations.

Our aim is to estimate the optimal topology for such problems when a novel loading condition is prescribed. For this, we generate a pool of training data where each training sample consists of a known loading configuration, and a corresponding optimal topology. To compute the optimal topologies given the loads, we use a density-based topology optimization algorithm given in (Andreassen, Clausen, et al. 2011). The method assigns a density value, ρ_e, between $[0, 1]$ to each pixel e in the domain that dictates the Young's modulus E_e for that particular pixel as:

$$E_e(\rho_e) = E_{min} + \rho_e^p(E - E_{min}) \tag{1}$$

where E is material stiffness, E_{min} is a very small stiffness value assigned to void regions and p is the penalization factor to attain black and white images. The optimization works toward minimizing the compliance resulting from the generated gray-scale structure. Mathematical formulation for the optimization procedure is given as follows:

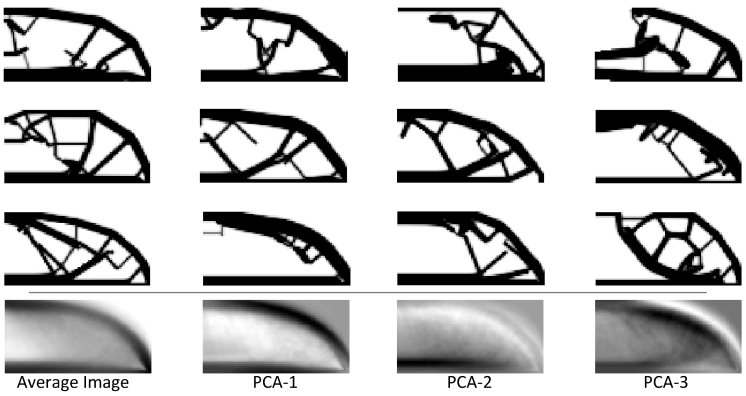

Figure 3.: Top: Example training samples. Bottom: Average image and first three PCA images.

$$\begin{aligned} \underset{\boldsymbol{\rho}}{\text{minimize}} \quad & c(\boldsymbol{\rho}) = \mathbf{U}^T\mathbf{K}\mathbf{U} \\ \text{subject to} \quad & V(\boldsymbol{\rho})/V_0 = r, \\ & \mathbf{K}\mathbf{U} = \mathbf{F}, \\ & 0 \leq \boldsymbol{\rho} \leq 1. \end{aligned} \tag{2}$$

where the objective function c is the compliance and $\mathbf{U}$, $\mathbf{K}$ and $\mathbf{F}$ are the global displacement vector, stiffness matrix and force vector, respectively. r is the predetermined fraction of material volume $V(\boldsymbol{\rho})$ and design space volume V_0. Details of this formulation can be found in (Andreassen, Clausen, et al. 2011).

As a result of the optimization process, resulting images with a varying Young's modulus field represent the optimal topologies for the corresponding loading configurations, where lighter colors represent weaker regions (lower stiffness). The collection of these images establish an input database to our method.

4. Component Analysis

Principal component analysis is useful for analyzing the input data to identify the significant features inherent in the data, as a way to facilitate dimensionality reduction with minimal information loss.

Suppose we have M images each with N_x-by-N_y resolution in our dataset. Then, gray-scale density values for images are stacked into M column vectors $\mathbf{t_j}$ $(j = 1, ..., M)$ of length $l = N_x \times N_y$ to form the high-dimensional image space feature vectors. To mean-shift the data, the average image $\bar{\mathbf{t}}$ is calculated and subtracted from each sample in the training dataset, i.e. $\mathbf{t_j} - \bar{\mathbf{t}}$. In Fig. 3, a randomly selected subset of the example training dataset (with 1000 samples), the resulting mean image, and the first three PCA component images are shown.

Let the mean centered image be $\mathbf{p_j} = \mathbf{t_j} - \bar{\mathbf{t}}$, we can store our entire data set into lxM matrix $\mathbf{P}$ to perform principal component analysis. Each column of $\mathbf{P}$ represents one mean centered image. In order to obtain the eigenvectors (i.e. principal components), the covariance matrix can then be constructed as $\mathbf{C} = \mathbf{P}\mathbf{P}^\mathbf{T}$. However, size of the matrix $\mathbf{C}$ is l-by-l and calculating l eigenvectors may not be practical. As mentioned in (Turk and Pentland 1991), if the number of features is larger than the number of training images ($l >> M$), there can be at most $(M - 1)$ useful eigenvectors (corresponding to non-zero eigenvalues) instead of l. These eigenvectors of l-by-l $\mathbf{P}\mathbf{P}^\mathbf{T}$ matrix can be determined from the eigenvectors of M-by-M matrix $\mathbf{P}^\mathbf{T}\mathbf{P}$ as $\mathbf{c_j} = \mathbf{P}\mathbf{v_j}$ where $\mathbf{v_j}$ is eigenvectors

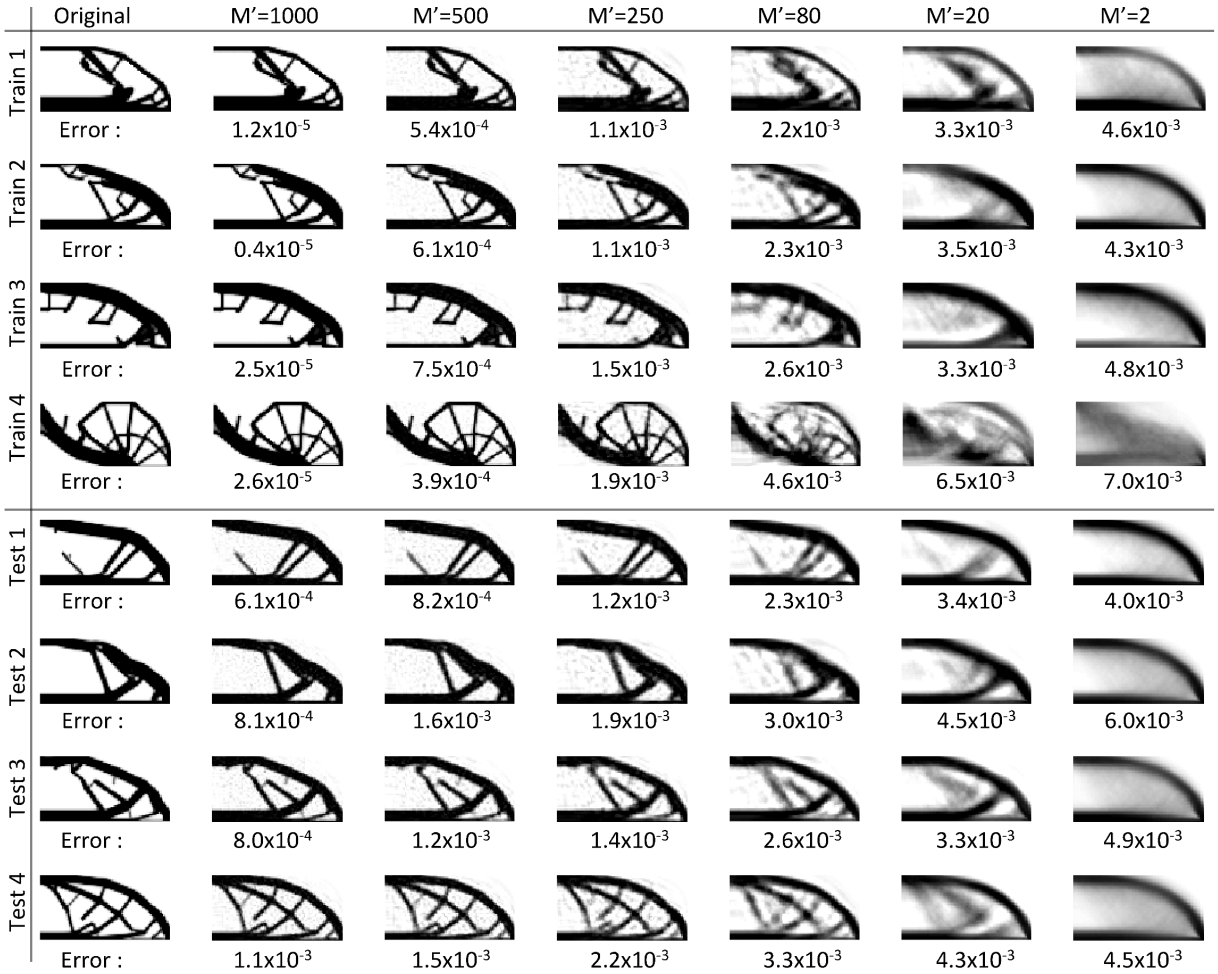

Figure 4.: Reconstruction of example samples relative to the different number of eigen-components used. Each row corresponds to a different example. The upper half illustrates reconstructions for sample training images. The lower half shows the same for test images (i.e., images not involved in the construction of PCA). Difference between each reconstruction image and the original one is evaluated using Eq. 3 and given underneath the corresponding image.

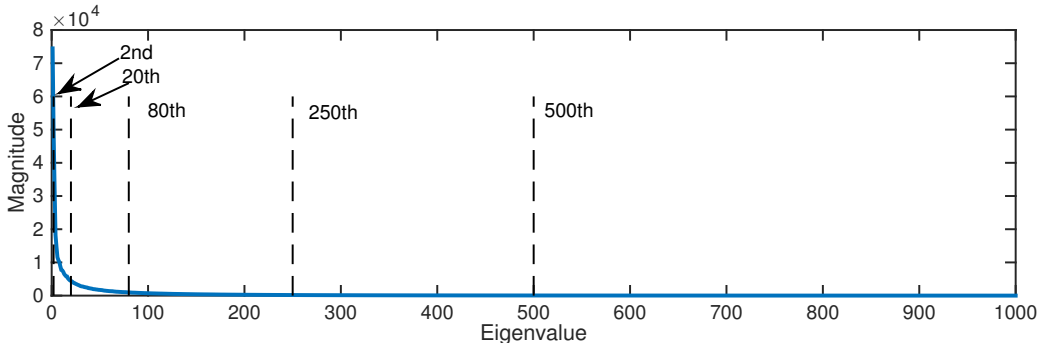

Figure 5.: Magnitudes of eigenvalues corresponding to non-zero eigen-images obtained with PCA.

of $\mathbf{P^T P}$. In this paper, $\mathbf{c_j}$'s will be referred to as eigen-images. Each input topology optimization image can then be represented as a linear combination of these M eigen-images, resulting in a PCA weight vector of $\mathbf{W_i} = [w_1, w_2, ..., w_M]^T$. Even using only a few number of eigen-images, M', associated with the largest eigenvalues, a good approximation of an image can be obtained.

In Fig. 4, reconstruction of sample topology optimization images with different number of eigen-

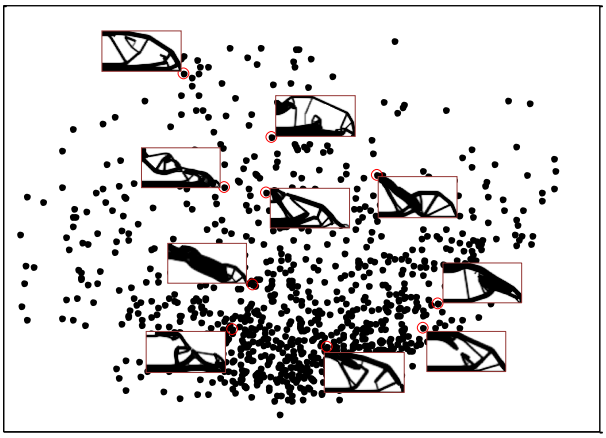

Figure 6.: Training images projected into the 2D space created by the first two PCA components.

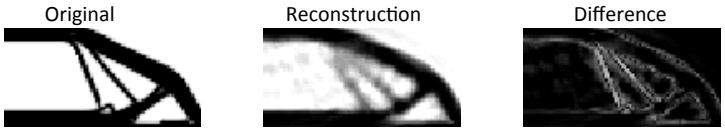

Figure 7.: Difference between an example topology and its reconstruction using first 80 eigen-images.

images are illustrated. Note that the figure shows example reconstructions of samples that were used during training (train), as well as for novel samples that were not part of the training (test). In this example, images are 80-by-40 pixels and there are 1000 training images generated by random assignments to the loading configurations and solving for the corresponding optimal topologies. Since PCA is limited by the number of training samples ($1000 < 3200$), the number of non-zero eigen-images is 1000. In Fig. 5, magnitudes of the eigenvalues corresponding to these eigen-images are shown. Here, it can be observed that a remarkably small number of eigen-images are sufficient for a high-fidelity reconstruction of the original images. Based on these observations, we use the first 80 eigen-images in our examples in remainder of the paper, without loss of generality.

Fig. 6 shows the dataset of 1000 training images when projected to the space created by the first two eigen-vectors.

To quantify the mismatch between an original image and its reconstruction, we use the L_1 distance between the two images:

$$d(t_1, t_2) = \frac{\|t_1 - t_2\|_{L1}}{length(t)} \tag{3}$$

Fig. 7 illustrates this difference. This metric provides a value between $[0, 1]$ where 0 represents identical images. For this particular example, the difference between the original image and its reconstruction is calculated as 2.8×10^{-3} using Eq. 3. Fig. 4 also demonstrates the difference values evaluated for various examples underneath each corresponding image.

When properly weighted, eigen-images can be linearly combined to create an approximation to a new image representing a new optimized topology. For this purpose, a set of PCA weights associated with the corresponding loading configuration should be estimated. We address this problem by introducing a mapping function between the loading configurations, $\mathbf{F}$ and the PCA weights, $\mathbf{W}$ of the training samples. Details of this process will be described in the next section.

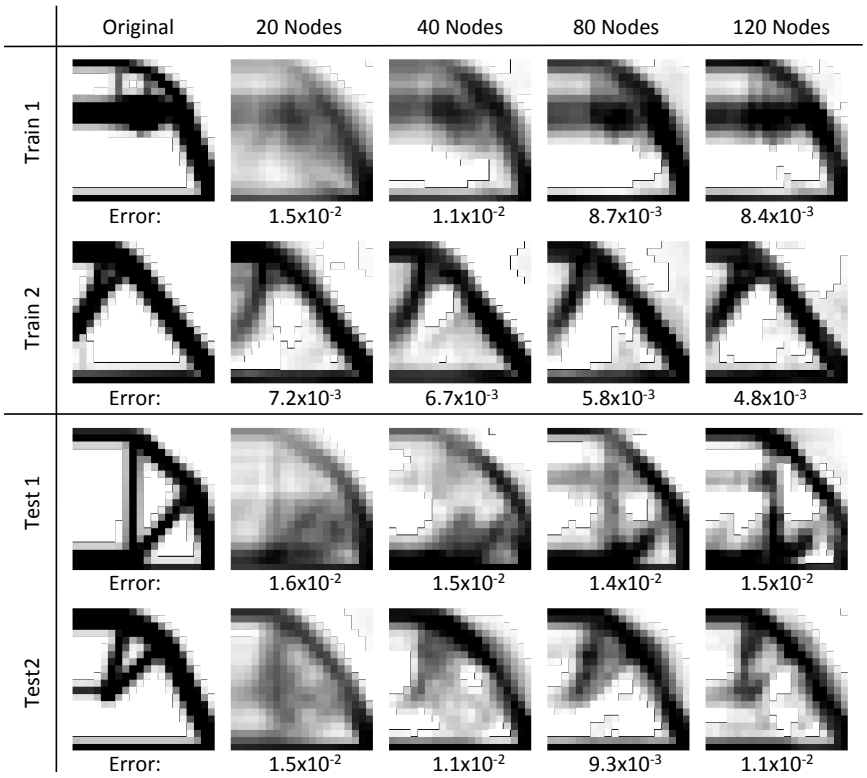

Figure 8.: Estimation performance of different neural network configurations with different number of hidden layer nodes. Each row corresponds to a different example. The upper half illustrates estimations for sample training images. The lower half shows the same for test images (i.e., images not involved in the training of neural networks). Difference between each estimation image and the original one is evaluated using Eq. 3 and given underneath the corresponding image.

5. Mapping Load Configurations to Optimal Topologies

A useful application of PCA decomposition is that with a low dimensional data, a mapping between the original input and the PCA vector space can be created. In this section, we present a neural network approach to generate this mapping, specifically between the force vector indicating load conditions ($\mathbf{F_i}$) and the PCA weights ($\mathbf{W_i}$). In order to show the effectiveness of the neural network mapping, we compare its performance with linear regression and polynomial regression approaches in the following section.

In our experiments, the mapping functions uses the following input and output configurations:

(1) The input vector is composed of four real numbers (x and y positions and magnitudes) for each force in the problem.
(2) The output vector is composed of 80 real numbers corresponding to the PCA weights.

Neural Network: Theoretically, a neural network with a sufficient number of nodes and training samples is able to learn any input-output relationship for regression. However, with high dimensional data, such regression would require a considerably large number of training samples and hidden layer nodes resulting in impractical convergence time in the training stage (Bishop and Nasrabadi 2006). We present results that indicate with limited amount of data and empirically determined number of hidden layer nodes, the neural network can appropriately learn the mapping from input force vectors to the output PCA weights for topology optimization.

In our experiments, we utilize a fully-connected feed-forward single hidden layer neural network (Bishop and Nasrabadi 2006) as the learner with the aforementioned input and output configurations. We train the resulting neural network with the scaled conjugate gradient algorithm. In order

to determine the number of nodes in the hidden layer, we conducted a set of experiments with different number of hidden layer nodes. In Fig. 8, a comparison of neural network performances are presented. In these experiments, neural networks are trained with the same dataset created for a specific design domain (middle configuration in Fig. 11). This dataset includes 400 training images of size 20-by-20 pixels. Each network is trained until the same level of convergence in network design variables (i.e. adaptive weights) is achieved. The performance of each neural network is then evaluated by using randomly selected samples from both the training and test dataset (a set of randomly generated samples that are not in the training dataset). Expectedly, as the number of hidden layer nodes increases, the neural network performs better for samples in the training dataset. However, the performance gets worse for test samples due to overfitting. Based on this experiment, we use 80 nodes in the hidden layer for all examples presented in the results section without loss of generality.

Linear Regression: As an alternative method, linear regression is one of the most commonly used approaches for learning the mapping between a set of feature vectors (Allen, Curless and Popović 2003). However, the relationship between the loading configurations and the PCA weights in such a high dimensional space may not be accurately predicted by this approach. In this paper, we present the linear regression for performance comparison purposes.

In order to construct the mapping between the input vector (i.e. loading configuration, $\mathbf{F}$) and output vector (i.e. PCA weights, $\mathbf{W}$), the mutivariate linear regression model can be formulated as follows:

$$\begin{aligned} \mathbf{W} &= \mathbf{F}\boldsymbol{\beta} \quad \text{where,} \\ \mathbf{W} &= [\mathbf{W_1}, \mathbf{W_2}, ..., \mathbf{W_i}, ..., \mathbf{W_N}]^T, \\ \mathbf{F} &= [\mathbf{F_1}, \mathbf{F_2}, ..., \mathbf{F_i}, ..., \mathbf{F_N}, 1]^T. \end{aligned} \tag{4}$$

Here, N is the number of samples in the training dataset (400 in our experiments), vector $\mathbf{W_i}$ includes M' PCA weights (80 in our experiments) and $\boldsymbol{\beta}$ represents the parameter matrix to be obtained. Eq. 4 can be solved as $\boldsymbol{\beta} = \mathbf{F}^{\dagger}\mathbf{W}$ where $\mathbf{F}^{\dagger}$ is pseudo-inverse of input matrix $\mathbf{F}$.

Polynomial Regression: The linear regression method formulated in Eq. 4 can be extended to polynomial regression in order to account for the non-linearity in the design space. Theoretically, it may be possible to model non-linear behavior in the design space with high order polynomials. However, computational cost increases as the order of the polynomial increases, especially for high dimensional spaces with a large number of training samples due to the (pseudo-)inverse calculation of a large matrix. In this paper, we use only quadratic regression for illustration purposes since we do not have any prior knowledge on the complexity of the design space. The mathematical formulation for the quadratic regression is given as:

$$\begin{aligned} \mathbf{W} &= \mathbf{Q}\boldsymbol{\beta} \quad \text{where,} \\ \mathbf{W} &= [\mathbf{W_1}, \mathbf{W_2}, ..., \mathbf{W_i}, ..., \mathbf{W_N}]^T, \\ \mathbf{Q} &= [\mathbf{Q_1}, \mathbf{Q_2}, ..., \mathbf{Q_i}, ..., \mathbf{Q_N}, 1]^T, \\ \mathbf{Q_i} &= [\mathbf{F_{i1}^2}, \mathbf{F_{i2}^2}, ..., \mathbf{F_{ik}^2}, \mathbf{F_{i1}F_{i2}}, \mathbf{F_{i1}F_{i3}}, ..., \mathbf{F_{i(k-1)}F_{ik}}, \mathbf{F_{i1}}, \mathbf{F_{i2}}, ..., \mathbf{F_{ik}}]. \end{aligned} \tag{5}$$

Here, $\mathbf{F_{ik}}$ is the k'th element of input vector $\mathbf{F_i}$. Different from Eq. 4, regressors in $\mathbf{Q}$ additionally include second order terms. Similar to linear regression, Eq. 5 can be solved as $\boldsymbol{\beta} = \mathbf{Q}^{\dagger}\mathbf{W}$ where $\mathbf{Q}^{\dagger}$ is the pseudo-inverse of input matrix $\mathbf{Q}$. As seen, when the order of polynomial increases, the size of the matrix to be inverted increases drastically. For example, when there is only one force in the design domain (represented by four elements in $\mathbf{F_i}$), the size of $\mathbf{F}$ is 400-by-5 for linear

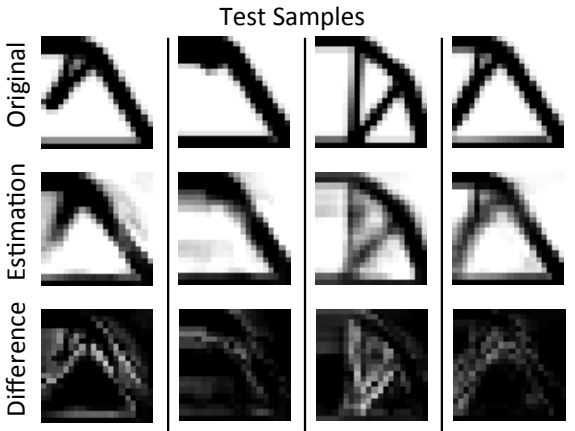

Image Difference (Error)		9.4×10^{-3}	5.6×10^{-3}	1.9×10^{-2}	8.0×10^{-3}
Objective Value, c	Original	27.8409	8.9304	14.8776	0.0339
	Estimation	38.3399	11.0153	24.8784	0.0455

Figure 9.: Example topologies and their neural network estimations using the first 80 eigen-images. Difference between original image and estimations (calculated using Eq. 3) and objective values (structural compliance) calculated using Eq. 2 are also shown for numerical comparison.

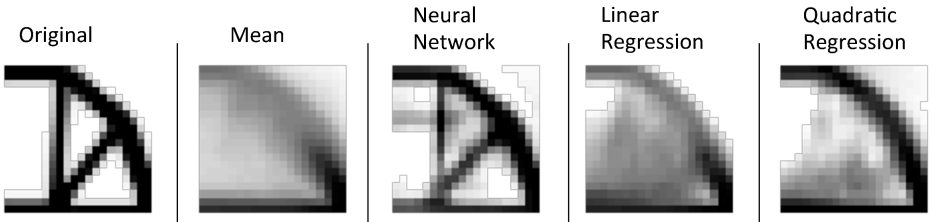

Image Difference (Error)		3.1×10^{-2}	2.3×10^{-2}	2.7×10^{-2}	3.0×10^{-2}
Objective Value, c	14.8776	55.7447	25.4613	40.2057	75.5033
Training time [s]	-	-	363.18	12.57×10^{-3}	17.04×10^{-3}
Estimation time [s]	-	-	0.39×10^{-3}	0.19×10^{-3}	0.23×10^{-3}

Figure 10.: Comparison of neural network estimation with linear regression and quadratic regression estimation of an example topology. Difference between original image and estimations (calculated using Eq. 3), objective values (structural compliance, calculated using Eq. 2), training and estimation times are also shown for numerical comparison.

regression with 400 training samples. However, the size of $\mathbf{Q}$ is 400-by-15 for quadratic regression. This difference is even more substantial for higher order polynomials, especially when the number of input forces or training samples are large.

6. Results and Discussions

With a sufficient number of training samples, the neural network can generate a precise mapping between the loading configurations and the corresponding PCA weights. However, the resulting estimation can be affected by the number of eigen-images used to express that image. Fig. 9 illustrates the performance of our approach on several test samples for a specific design domain (middle

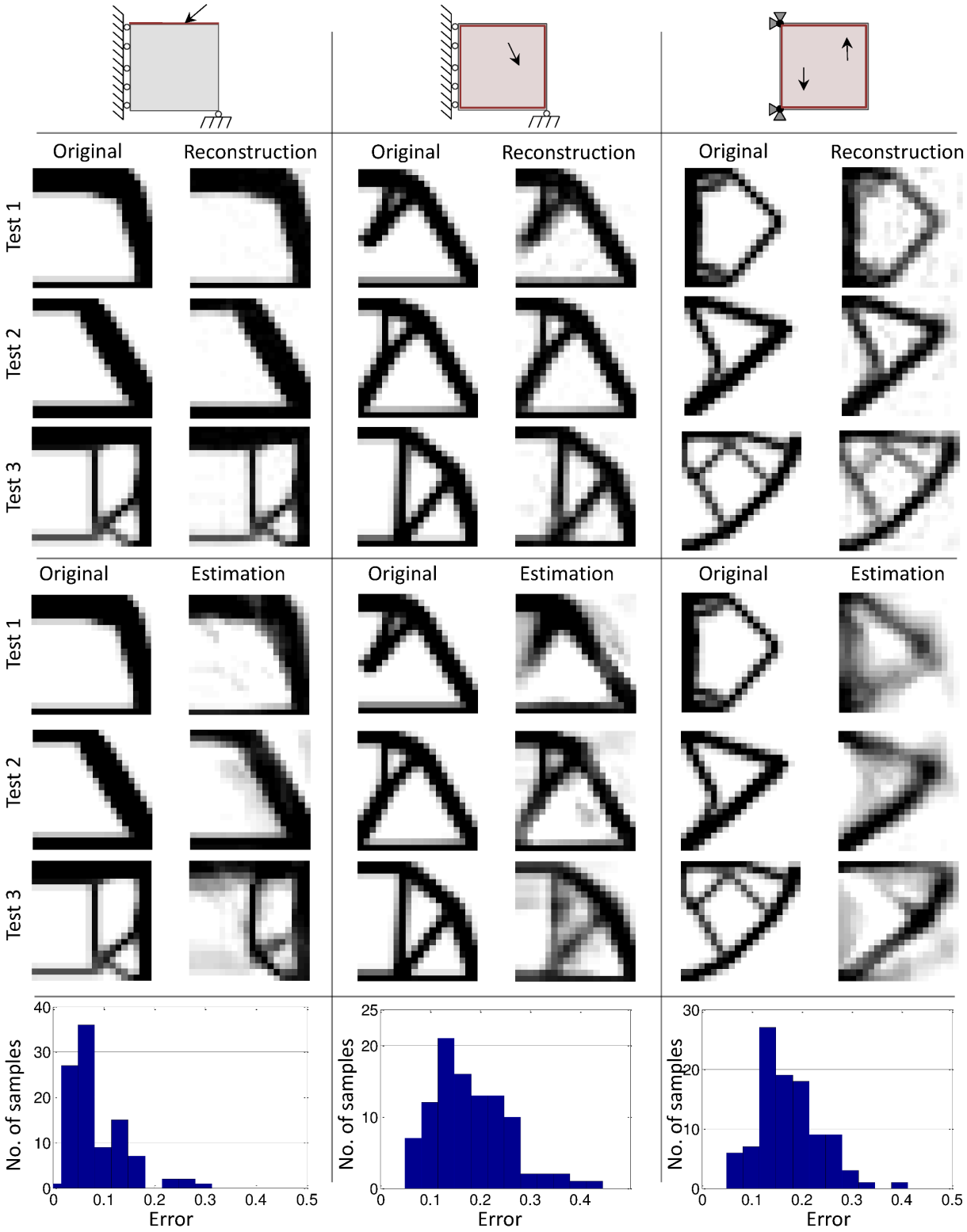

Figure 11.: Performance of our method for different design configurations. Left: single force anywhere on the top surface in any direction and magnitude. Middle: single force anywhere in the domain in any direction and magnitude. Right: two vertical forces anywhere in the domain in any magnitude. Reconstruction refers to the PCA reconstruction of the sample using the eigen-images. Estimation uses the proposed neural network, followed by PCA reconstruction.

Sample	Configuration		
	Left	**Middle**	**Right**
Test 1	0.30×10^{-3} s	0.34×10^{-3} s	0.36×10^{-3} s
Test 2	0.41×10^{-3} s	0.39×10^{-3} s	0.30×10^{-3} s
Test 3	0.38×10^{-3} s	0.34×10^{-3} s	0.29×10^{-3} s
Training Time	485.09 s	363.18 s	625.19 s

Table 1.: Computation times for the tests cases demonstrated in Fig. 11.

configuration in Fig. 11). In this configuration, the PCA and neural network are trained using 400 samples and tested with randomly generated loading configurations. As previously mentioned, only the first 80 eigen-vectors are used for reconstruction and estimation. Fig. 9 shows the original samples and their corresponding estimations using our method. Note that estimation involves using the neural network to map the loading configuration into the PCA weight vector, followed by a PCA based reconstruction using 80 samples. Besides the image differences, compliance value given in Eq. 2 can also be compared to evaluate the performance. In Fig. 9, the table shows the compliance values obtained for regular topology optimization result and our estimation together with the image differences. Here, it can be observed that image difference directly reflects the percent change in the compliance value. Hence, one can judge the performance by only checking the image difference computed with our metric given in Eq. 3.

In Fig. 10, we compare the performance of neural network versus linear regression and quadratic regression formulated in Eq. 4 and Eq. 5 on a sample test image. All of the mapping methods are trained with the same 400 samples. Note that linear and quadratic regression fail to reproduce some of the details in the optimal topology. It can also be observed that the quadratic regression performs even worse than linear regression for this specific test case. Fig. 10 also demonstrates the numerical values for image differences (from the original), compliance values and computation times for training and estimation. Although neural network performs significantly better than the other methods, there is a time trade-off because of the training process. The training and testing of each mapping method is conducted on a PC with a 2.4GHz Core CPU and 8GB RAM using MATLAB R2014b.

Fig. 11 illustrates the performance of our algorithm on several test configurations. In the top row, basic representations of the design problems are illustrated. Here, boundary conditions and loading configurations are shown. Red areas represent spaces where forces can be placed. In the following row, reconstructions of several test samples using eigen-images are presented. Since only the first 80 eigen-images are used to construct an image, there are slight differences from the original optimal topologies. We compare our estimation results with the optimal topologies in the third row. As the design problem becomes more complex, the accuracy of resulting estimations reduces since the number of training samples for neural network may fail to be sufficient. Better estimations can be made using a higher number of training samples for more complex problems. In the last row of Fig. 11, a histogram showing the distribution of error between the optimal topology and the estimation result of our approach among 100 test samples is given for each design configuration. Although we use a limited number of training samples and PCA components, the main structure for optimal topologies can be estimated. Computation times obtained for the same test samples are shown in Table 1 together with the neural network training times.

In design problems involving complex loading configurations, the reconstruction accuracy may decrease visually (e.g., estimations in the last column of Fig. 11). However, even in those cases, our optimal topology estimates can be used as an initial condition for a conventional topology optimization algorithm, e.g. (Andreassen, Clausen, et al. 2011; Sigmund 2001), to reduce the convergence time. Fig. 12 shows the effect of using our estimation as an initial condition. For around 70% of 100 test samples, reduction in convergence time is observed for the posed problems. This gain can be even more significant for larger and more complex design domains.

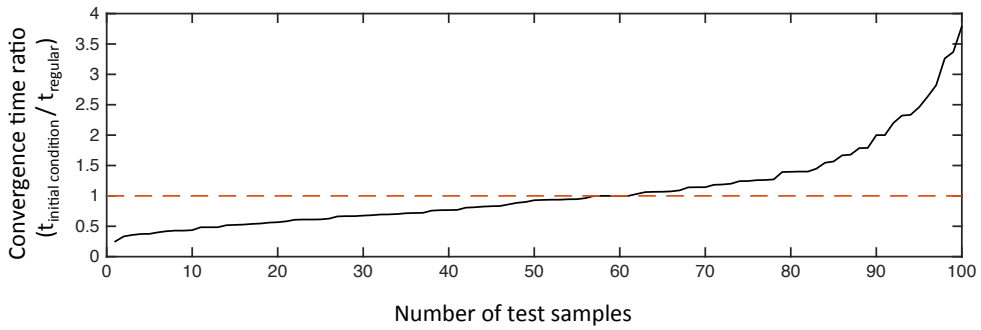

Figure 12.: Convergence time comparison. Time of convergence when neural network estimation is used as initial condition ($t_{initial\ condition}$) is shorter than that of regular topology optimization ($t_{regular}$) for most of the test samples. Configuration used: Fig. 11 (left).

7. Conclusions

We explore the feasibility and performance of a data-driven approach to topology optimization problems involving structural mechanics. We take a set of optimal topology examples for a given configuration, and project them into a lower dimensional space with PCA analysis. We then learn a mapping from loading configurations to optimal topologies using neural networks. Using the trained network, we studied the performance of estimating optimal topologies for novel loading configurations. Our results show that the proposed method can successfully predict the optimal topologies in different problem settings. Moreover, we also prove that the topologies predicted by the proposed method are effective initial conditions for faster convergence in subsequent topology optimization. We believe such time and computational power savings will be greater as the problem size and complexity increase. Thus, a valuable future direction is the application of the proposed method for 3D topology optimization.

References

Aage N, Lazarov BS. 2013. Parallel Framework for Topology Optimization Using the Method of Moving Asymptotes. Structural and Multidisciplinary Optimization. 47(4):493–505.

Allen B, Curless B, Popović Z. 2003. The Space of Human Body Shapes: Reconstruction and Parameterization from Range Scans. ACM Transactions on Graphics. 22(3):587–594.

Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O. 2011. Efficient Topology Optimization in Matlab Using 88 Lines of Code. Structural and Multidisciplinary Optimization. 43:1–16.

Bendsoe MP. 1989. Optimal Shape Design as a Material Distribution Problem. Structural Optimization. 1(4):193–202.

Bendsoe MP, Kikuchi N. 1988. Generating Optimal Topologies in Structural Design Using a Homogenization Method. Computer Methods in Applied Mechanics and Engineering. 71(2):197–224.

Bendsoe MP, Sigmund O. 2004. Topology optimization: Theory, methods and applications. 2nd ed. Springer.

Bishop CM, Nasrabadi NM. 2006. Pattern recognition and machine learning. vol. 1. springer New York.

Blanz V, Vetter T. 1999. A morphable model for the synthesis of 3d faces. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co.; p. 187–194.

Chapman CD, Saitou K, Jakiela MJ. 1994. Genetic Algorithms as an Approach to Configuration and Topology Design. Journal of Mechanical Design. 116:1005–1012.

Cox T, Cox M. 1994. Multidimensional scaling. London: Chapman and Hall.

Dijk NP, Maute K, Langelaar M, Keulen F. 2013. Level-Set Methods for Structural Topology Optimization: A Review. Structural and Multidisciplinary Optimization. 48(3):437–472.

Guest JK, Smith Genut LC. 2010. Reducing dimensionality in topology optimization using adaptive design variable fields. International Journal for Numerical Methods in Engineering. 81(8):1019–1045.

Hassani B, Hinton E. 1998. A Review of Homogenization and Topology Optimization IHomogenization Theory for Media with Periodic Structure. Computers and Structures. 69(6):707–717.

Jakiela MJ, Chapman C, Duda J, Adewuya A, Saitou K. 2000. Continuum Structural Topology Design with Genetic Algorithms. Computer Methods in Applied Mechanics and Engineering. 186(24):339 – 356.

Jolliffe I. 2005. Principal component analysis. Wiley Online Library.

Kirby M, Sirovich L. 1990. Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces. IEEE Transactions on Pattern Analysis and Machine Intelligence. 12(1):103–108.

Norato JA, Bendsoe MP, Haber RB, Tortorelli DA. 2007. A Topological Derivative Method for Topology Optimization. Structural and Multidisciplinary Optimization. 33(4-5):375–386.

Richardson JN, Coelho RF, Adriaenssens S. 2010. Robust topology optimization of 2d and 3d continuum and truss structures using a spectral stochastic finite element method. In: 10th World Congress on Structural and Multidisciplinary Optimization (WCSMO˜ 10); May.

Roweis ST, Saul LK. 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science. 290(5500):2323–2326.

Rozvany GIN. 2001. Aims, Scope, Methods, History and Unified Terminology of Computer-Aided Topology Optimization in Structural Mechanics. Structural and Multidisciplinary Optimization. 21(2):90–108.

Rozvany GIN. 2009. A Critical Review of Established Methods of Structural Topology Optimization. Structural and Multidisciplinary Optimization. 37(3):217–237.

Rozvany GIN, Zhou M, Birker T. 1992. Generalized Shape Optimization without Homogenization. Structural Optimization. 4(3-4):250–252.

Schramm U, Zhou M. 2006. Recent developments in the commercial implementation of topology optimization. In: Bendsoe MP, Olhoff N, Sigmund O, editors. IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials; Solid Mechanics and Its Applications; vol. **137**. Springer Netherlands; p. 239–248.

Sethian J, Wiegmann A. 2000. Structural Boundary Design via Level Set and Immersed Interface Methods. Journal of Computational Physics. 163(2):489–528.

Sigmund O. 2001. A 99 Line Topology Optimization Code Written in MATLAB. Structural and Multidisciplinary Optimization. 21(2):120–127.

Sirovich L, Kirby M. 1987. Low-Dimensional Procedure for the Characterization of Human Faces. Journal of Optical Society of America A. 4(3):519–524.

Suzuki K, Kikuchi N. 1991. A Homogenization Method for Shape and Topology Optimization. Computer Methods in Applied Mechanics and Engineering. 93(3):291–318.

Tenenbaum J. 1998. In: Jordan M, Kearns M, Solla S, editors. Advances in Neural Information Processing 10. Cambridge, MA: MIT Press; p. 682–688.

Turk M, Pentland A. 1991. Eigenfaces for Recognition. Journal of Cognitive Neuroscience. 3(1):71–86.

Wang MY, Wang X, Guo D. 2003. A Level Set Method for Structural Topology Optimization. Computer Methods in Applied Mechanics and Engineering. 192(12):227–246.

Yumer ME, Kara LB. 2011. Conceptual design of freeform surfaces from unstructured point sets using neural network regression. In: Proceedings of ASME International Design Engineering Technical Conferences/DAC. ASME.

Yumer ME, Kara LB. 2012. Surface Creation on Unstructured Point Sets Using Neural Networks. Computer-Aided Design. 44(7):644–656.