Data Driven Design: Exploring Structural Forms using Machine Learning and Graphic Static

Fuhrimann Lukas

July 4, 2018

Master Thesis

ETH Zürich, Chair of Structural Design, Prof. Dr. Joseph Schwartz

Supervised by:
Patrick Ole Ohlbrock
Pierluigi D'Acunto
Vahid Moosavi

Abstract

The objective of this thesis is to develop a design supporting approach that combines the strength of human designers and the power of computational methods. In particular, this work aims at inspiring designers with structurally informed geometries that go beyond typological boundaries and their imagination. The thesis combines various techniques from two different research fields. On the one hand, an equilibrium network generator, Combinatorial Equilibrium Modeling, that is able to transform a given system of information (topology, metrics) into a spatial geometry in equilibrium. On the other hand, machine-learning techniques, Self Organizing Map and Uniform Manifold Approximation and Projection, which enable the organization and analysis of huge amounts of these equilibrium geometries simultaneously. One of the core parts of the project deals with the representational bridge between these two fields. Moreover, various different sub-algorithms and methods such as deep auto-encoders and generative adversarial networks are combined, tested and evaluated with respect to their usefulness for the design phase. As a result, an interactive hierarchical clustering methodology is developed, which enables the representation of topologically and spatially similar equilibrium networks in a well organized 3-dimensional map. This map of equilibrium structures is in a next step studied as an inspiration for an undefined design brief. The geometrical and structural information of a selected design is finally adopted for the construction of a physical sculpture.

Acknowledgements

A team of brilliant researchers and designers: Ole Ohlbrock, Pierlugi D'Acunto and Vahid Moosavi supported this project from scratch to the end. I would like to thank these three people for the enthusiasm, the guidance and the know-how they provided to make this thesis possible.

Further, I would like to thank my parents: Stephan and Ursula, for pushing me to get the great education I received at ETH. Always supporting me and providing all the mental and financial means to pursue my passion.

Last but not least a big thanks to Professor Joseph Schwartz for providing the setup for this thesis. And all the researchers from the ITA who made this last year a wonderful time at ETH and Zürich.

Table of Contents

1 Introduction to Structurally Informed Form Exploration

In the early design phase, the task of Architects and Engineers usually involves to think out of the box, to come up with diverse and innovative solutions. Scientific observations have shown that it is crucial to consider a variety of meaningful but diverse solutions, in order to grasp the whole solution space (all the possible solutions) and ultimately reach a satisfying design.

"However, designers do not always find it easy to generate a range of alternative solutions in order that they better understand the problem. Their ordering principles or primary generators can, of course, be found to be inappropriate, but designers often try to hang on to them, because of the difficulties of going back and starting afresh." Nigel Cross 2006, [1]

Various different approaches exist to create new designs e.g. imitate natural behavior [2], using geometrical patterns e.g. fractals [3] or geometrical rules such as shape grammars [4]. This research sets a focus on physically informed designs and specifically the exploration of structurally informed systems (Figure 1). These systems describe spatial designs informed by their load bearing behavior which allows to design from a structural perspective.

Heinz Isler and Frei Otto, two influential figures in the field of structural design, tried to overcome the difficulties described by Nigel Cross. Heinz Isler used "small" physical models to mimic natural phenomena in order to find a variety of funicular geometries that were then used as a blue-print for real-scale shell structures. One could argue that Isler was manually exploring big areas of the whole design space of compression shells within his career (Isler [5]). Frei Otto used similar approaches, but rather exploring light-weight structures that partially also involved structures with mixed compression tension systems (Frei Otto [6]). Both of these successful protagonists were bound to physical models for their design space exploration. Producing these models, extrapolating/upscaling the geometry is time consuming and liable to mistakes. To support designers in the generation of new and alternative structurally informed designs, a faster and more convenient method is desirable.

The increasingly important role of computers and numerical mathematics in the 20th century introduced a vast amount of tools. These tools are tailored to find optimized forms, to investigate their structural behavior, and also to visualize the latter. However, most of the algorithms and tools (Finite Element Method , Force Density Method, Dynamic Relaxation Method) support the designer in a quasi analytic procedure, in which many things have to be already clearly formulated. Most of the methods require information about material, geometric properties and topology. These information are then translated into a singular solution which altogether makes them generally inappropriate for an open design phase. Hence, applications of tools that aim to support the designer in the early design phase with a more open setup and the possibility of generating many comprehensible but unexpected and diverse solutions in a fast manner are generally missing.

Whereas the end of the 20th century introduced computer-aided design, the beginning of the 21st century exhibits new streams that connect Data-driven approaches, machine-learning algorithms and artificial intelligence to the world of design. Combining these Data-driven approaches with computational form generating tools opens up a promising field of research. Hence, a new way of investigating infinite design spaces of structural informed systems can be imagined. These solution spaces are not bound anymore to the slow process of building physical models (Heinz Isler and Frei Otto), but only to the computational setup.

First data-driven approaches supporting the idea of diversity-driven design (e.g. Brown and Mueller [7], Thornton Tomasetti (citation), Mueller and Ochsendorf [8] or Shea and Cagan [9]) have shown their great potential for structural design. However, these data-driven approaches rely on a rather typological approach, in which for a pre-classified structure (e.g. a beam/truss) different design variations with little diversity are investigated. Such an approach is very limited in its surprise and mainly useful for optimization tasks.

The present work, on the contrary, aims to develop a non-typological method for data-driven structural design exploration (Figure 1, right). Similar to the work of Juney Lee et. al. [10], this is achieved by generating unclassified structures (including the variation of topology) with a big diversity and therefore not narrowing the expected solution spaces. A methodology for the guided exploration of these spaces is then described by a qualitative study of different clustering methods.

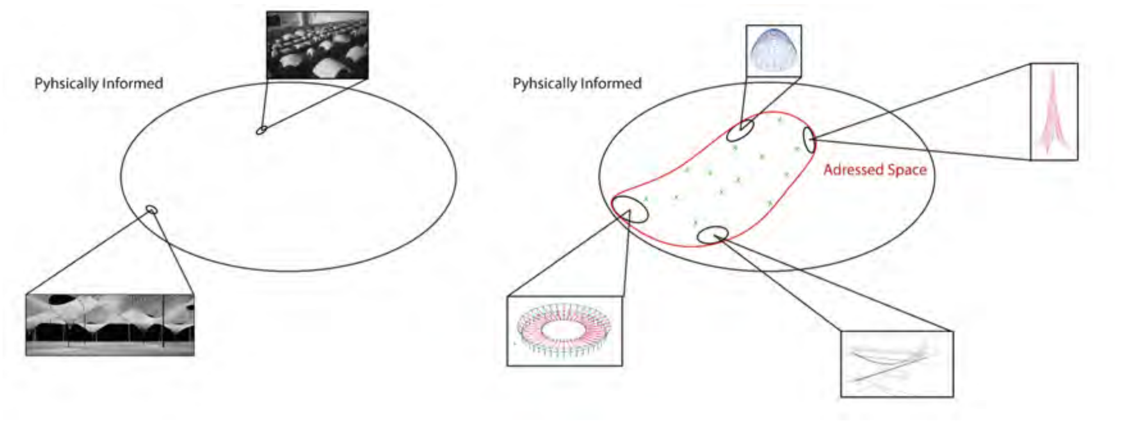

Figure 1: Physical Informed Design Space Exploration, shifting from Physical Models (left) and the Exploration of Typologies to an Automated Topological Design Space Exploration (right).

Data-driven approaches rely on data. The data-set that describes a solution space for structural design in the conceptual design phase should contain on the one hand, only structurally meaningful geometries (networks in equilibrium) and on the other hand also geometries that go beyond the known solutions, beyond the imagination of the designer. Combinatorial Equilibrium Modeling (CEM) can fulfill both requirements. CEM is a novel design approach based on graphic statics and graph theory. The CEM algorithm is able to quickly translate any ordered system of information (topology, metrics) into an equilibrium network. Thus it is used to generate diverse data sets that are representing the solution spaces. Automatizing the generation of equilibrium networks with CEM and using Machine-learning algorithms to find ways to understand the generated solution space, is one of the main contributions of this thesis. It shows how the task of the designer could cognitively shift from designing the artifact itself to the design of the system (Figure 2).

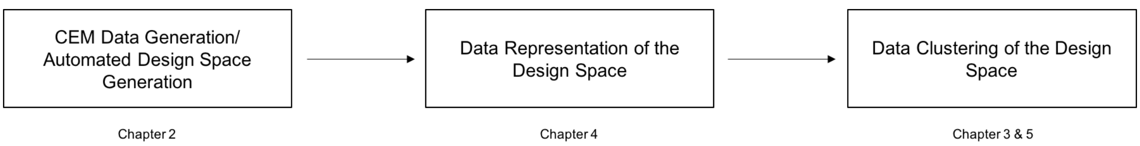

Figure 2: Design of the Generating and Clustering System.

2 Structural Form finding and Data Generation

Exploring a diverse solution space requires data as its primary source. In order to create structurally informed geometries, a tool that is based on physical laws and that allows a broad variation of solutions (tension/compression and hybrid structures) is essential. Combinatorial Equilibrium Modeling (CEM), a method for the design of spatial equilibrated networks, based on graphic statics, is used for the data generation within this work. Exploring infinite design spaces with an automated data generation is transforming the focus from the design of a singular artifact to the design of the generator-system. Designing the automatic generation within the CEM requires an explicit design methodology on a more abstract level. The goal of this work is to formulate a method that is not restricting the solution space to predefined and expected solutions but also presents as many good solutions as possible.

2.1 Combinatorial Equilibrium Modeling

Combinatorial equilibrium modeling is a method for the design of spatial networks in equilibrium that is based on graphic statics. This allows the generation of a design space with infinite different equilibrium states that can be conveniently represented using a form diagram F and a force diagram F* (Ohlbrock et al. [11]). Other than the conventional form and force diagrams, CEM introduces a generative topological diagram T, which enables the control of the load-bearing behavior of a network in equilibrium by varying the connectivity of the network, its combinatorial state (tension or compression), the length of its elements as well as the magnitudes of its inner forces. Within the CEM framework, the members along the shortest paths are defined as trail members while the others are defined as deviation members. Start vertices s are those vertices of a trail with the maximum topological distance w from their corresponding support vertices. The topological diagram is split into layers k according to the topological distance w of the individual vertices to their corresponding support vertices. (Ohlbrock and Schwartz [12]).

Figure 3 shows the three different levels on which the interaction with the model can be described. The first level variations (topological variations) describe the connectivity of the initial topology. Within the second level variations (intermedi-

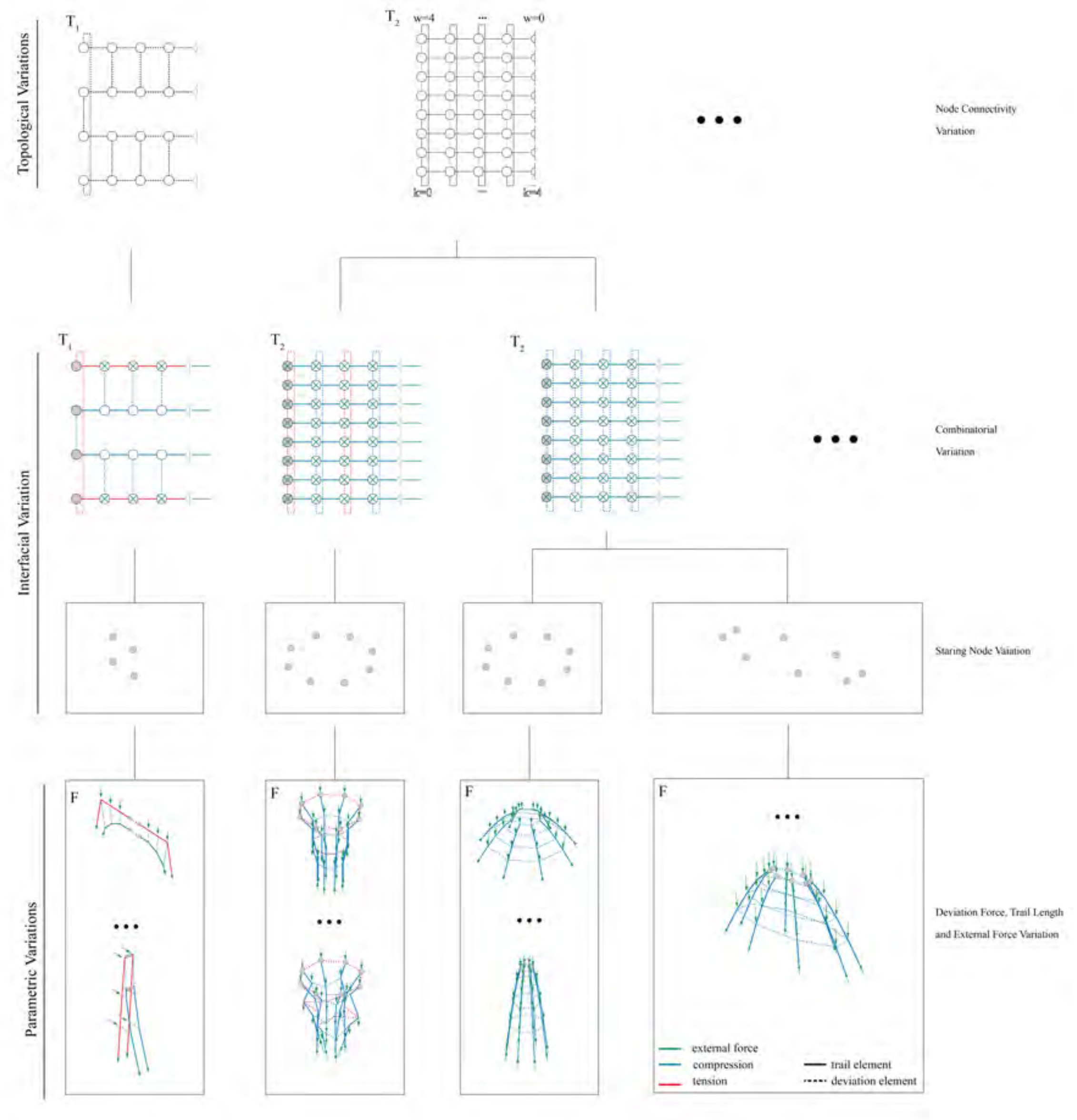

Figure 3: Topological, Interfacial and Parametric variations with CEM.

ate variations), the coordinates of the starting nodes and the combinatorial state (compression,tension) of the connections are assigned. The third level variations (parametric variations) describe the metric properties of the members and the external forces acting on the structure.

For the examples in Figure 3 an initial topological configuration consisting of 40 nodes, introducing 780 degrees of freedom which describe the possible connections between any two points, is chosen. Two topological variations, which represent different connectivities of the nodes are presented (top). The topological configuration on the right side of Figure 3 allows 88 degrees of freedom for its intermediate variation. 64 parameters assigning compression/tension members(second row from the Top) and 24 parameters defining the position of the starting nodes (third row from the Top). The parametric variation for latter comes with further 184 degrees of freedom, 32 parameters defining the inner force magnitudes of the deviation members, 32 parameters defining the lengths of the trail members and 120 parameters defining the external forces (bottom row).

A human designer can benefit from this method by manually exploring a single system. On the contrary, when a collection of systems should be considered, or the complexity of the system is increased, the design space gets more difficult to asses and predict. To be able to address the entire solution space (the red marked space in Figure 1), an automated design approach is needed.

2.2 Design Space Initialization

In order to automatize the generation of diverse equilibrium networks, explicit methods on all the three different levels are developed within this thesis. More specifically, the design space in this work is initialized with a parametric and a topological variation methodology. The topological variation involves the topological as well as the intermediate realm whereas the parametric focuses only on the metric variations. Based on these design space methods, it should be possible to carry out an abstract study of intriguing equilibrium forms as well as an exploration with a more pre-defined design brief.

2.2.1 Topological Design Space Initialization

The topological diagram introduced in section 2.1 can be interpreted as a graph. A graph is defined as an ordered pair of vertices's and edges. There exist various ways how to represent such graphs e.g. maximal, direct, over-direct, inverted graphs (Wortmann [13]). These algorithms seem to be useful to describe shapes, but in order to consider the sequential order of the CEM, a different approach is considered. A Dijkstra shortest path search algorithm is defining the sequential order of the topological elements from the defined starting node to the support. The edges along the shortest paths are defined as trail members (defined by length), the others either deviation or bracing members (defined by forces). To create an absolute definition of each member for all variations, a grid of vertices is predefined and can be altered by the creation and deletion of edges between the vertices. This allows to distinctly predict the members properties when altering the graph.

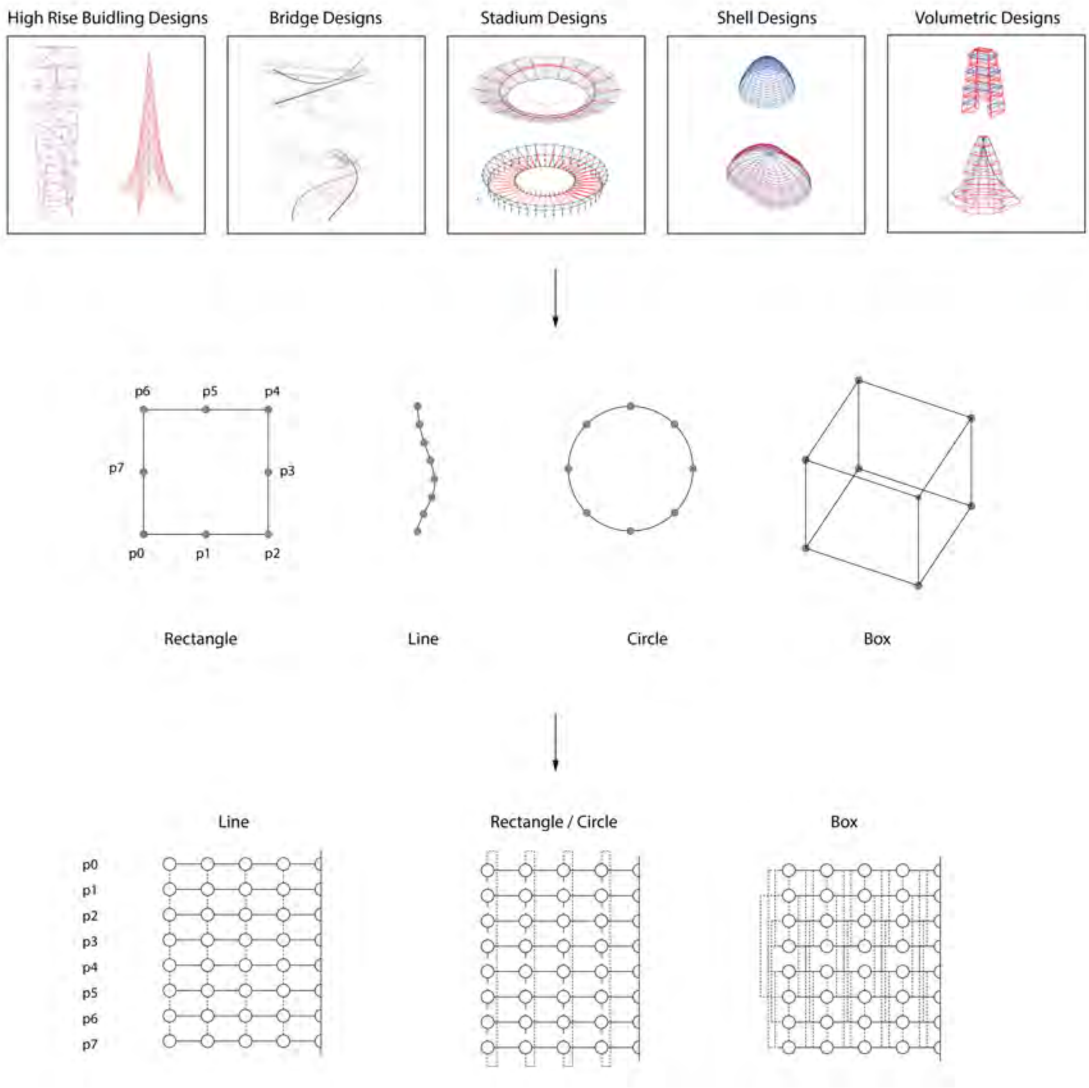

Figure 4: Starting Node and Initial Edge Connectivity configurations based on Former Design Studies.

To find informative and adequate equilibrium systems with CEM, the topological variations are defined by altering the initial topology with patterns. The initial topology consists of a defined number of 16 start vertices s and 9 (layers k). Further, initial configuration classes are defined considering a study of former design projects. The top row of Figure 4 shows examples of such former designs. Tanadini [14] describes the high rise building in his research with rectangular and linear starting node configurations. The bridge designs in the work of Biancardi [15] again follow a rectangular starting node configuration. For the stadium designs [11] of Ohlbrock, circular or elliptical node assemblies were implemented. The same configuration is also useful for the shell designs described within the parametric design space exploration. In order to force volumetric networks to be found in the solution space, the initial rectangular configuration can be expanded by one dimension to allow initial box/cube assemblies. These four configuration classes (Figure 4 middle) define the initial connectivity of nodes (Figure 4 bottom) and the initial starting node configuration.

Pattern	**Member**	**Description**
Layer number	All	Delete all possible connections to the layers that exist in a higher order than the layer number
Trail number	All	Delete all possible connections to the trails that exist in a higher order than the trail number
Delete	Deviation	Delete a single deviation member
Mirror X-axis	Deviation	Mirror previously deleted deviation member on X-axis
Mirror Y-axis	Deviation	Mirror previously deleted deviation member on Y-axis
Repeat along X-axis	Deviation	Repeat previous sequence of deleted deviation members along X-axis
Repeat along Y-axis	Deviation	Repeat previous sequence of deleted deviation members along Y-axis
Delete	Trail	Delete first trail member of a specified trail
Mirror X-axis	Trail	Delete opponent of previous deleted trail on X-axis
Mirror Y-axis	Trail	Delete opponent of previous deleted trail on Y-axis
Repeat along X-axis	Trail	Repeat the deletion of previous deleted trail member along X-axis
Repeat along Y-axis	Trail	Repeat deletion of previous deleted trail along Y-axis
Create	Bracing	Create bracing between two specified points
Mirror X-axis	Bracing	Mirror previous created bracings along X-axis
Mirror Y-axis	Bracing	Mirror previous created bracings along Y-axis
Repeat along X-axis	Bracing	Repeat previous created bracings along X-axis
Repeat along Y-axis	Bracing	Repeat previous created bracings along Y-axis
Fan Y-direction	Bracings	Connect a point with all points within neighboring layer
Fan X-direction	Bracings	Connect a point with all points within neighboring trail

Table 1: Table of applied patterns for topological variations.

The initial edge connectivity of the topology is related to the initial configuration class (box, rectangle, circle, line) which is described in Figure 4. Applying the patterns described in table 1, this initial connectivity can be altered and combined in a systematic and diverse way.

Next to topological variations, interfacial alterations are performed within the topological exploration. First, combinatorial configurations are assigned. Patterns which describe these configurations are defined in table 2 and can be combined in different setups.

Pattern	**Description**
alternate trail	Within a trail there is no variation
alternate layer	Within a layer there is no variation
Constant	The entire topology is either in compression or tension
Individual	each member is regarded separately

Table 2: Table of applied patterns for interfacial combinatorial variations.

Lastly, the position vectors of the starting nodes are varied within this step. The initial configuration class is defining a first configuration of the nodes which is then altered by changing the parameters that describe these initial classes. On top of the edge length alterations (b,l or h) of the line, rectangle and the box configuration, a sinusoidal function can be applied $y = A + B * sin(C * \phi)$ where $\phi \in \{0, 2\pi\}$ in order to describe configurations out of the axis of these edges.

Initial Configuration	**Description**
line	$x = l/i$ where i is the number of nodes in direction l and
Rectangle	$x = l/i$, $y = b/i$ where i is the number of nodes in direction l or b
Circle	$x = a * cos(\phi)$, $y = b * sin(\phi)$ where $\phi \in \{0, 2\pi\}$
Box	$x = l/i$, $y = b/i$, $z = h/i$ where i is the number of nodes in direction l, b, or h

Table 3: Table of parameters changing the initial starting node configuration.

2.2.1.1 Combinatorial Explosion

To understand the size of the solution space for the topological design exploration, all the possible solutions which result from topological and interfacial variations are exemplary calculated for an initial topological setup of 144 nodes:

- $x = n\frac{(n-1)}{2} = 144\frac{(144-1)}{2} = 10296$, possible single connections
- $10296! = 2.211398633 * 10^{36845}$, connection configurations
- $2.211398633 * 10^{36845}!$, interfacial combinatorial configurations
- $x^{48}!$ where $x \epsilon \mathbb{R}$, interfacial starting node configurations

For each of these combinations a metric value for the force or the length of each member and 144 values for the external forces has to be assigned. Within the topological design space exploration, these metric values are kept constant. A methodology on how these metric values could be varied in a systematic way is introduced in the parametric design space exploration in the next paragraph.

2.2.2 Parametric Design Space Initialization

The parametric design space initialization deals with a fixed topology and a constant initial starting node configuration. Varying the metrics of the deviation member forces, the length of the trails and the external forces, different equilibrium networks can be created. In order to reduce the number of independent parameters and therefore obtain a more systematic control over the generated spatial networks in equilibrium, relationships between input parameters are established by means of mathematical functions.

The herein investigated parametric space exploration is based on the topological diagram and starting node configuration of the third setup from the left in Figure 3, with an expanded number of 400 trail and 400 deviation members. For this topology, an equilibrium state can be obtained once a deviation force magnitude D is assigned to each deviation member and a constraint plane P is defined for each node (Ohlbrock et al. [11]). These planes fix the position of the nodes of the structure in space and thus control the lengths of the trail members. External loads, which represent the self-weight of the structure, are also applied at every node of the network. The positions of the start vertices s are fixed and equally distributed

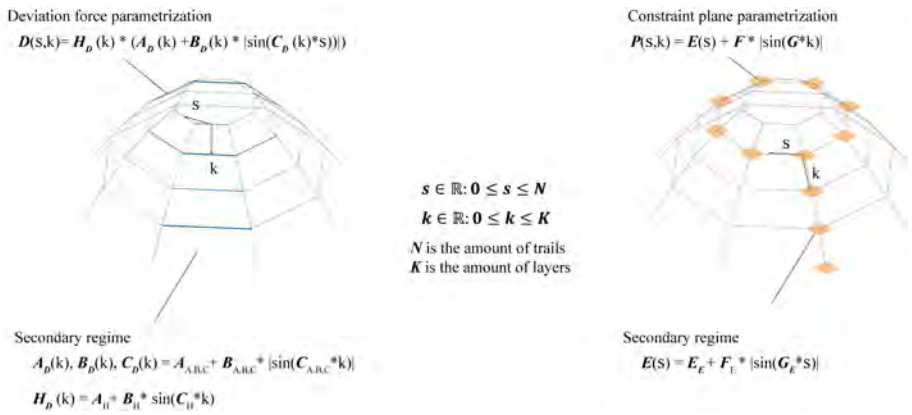

Figure 5: Parametric Set-Up.

on a circle with a variable radius R. This topological setup thus results in a total amount of 400 parameters describing the deviation forces D, 420 parameters for the constraint planes P and one parameter describing the radius R. In order to reduce the number of parameters and therefore obtain a more systematic control over the generated spatial networks in equilibrium, relationships between input parameters are established by sinusoidal functions are used to describe the relationship between parameters within the same layer k of the structure, but also to define the correlation between amplitudes over the various layers (Figure 5). Sinusoidal functions are chosen due to their continuity and smoothness. The parameters AA, AB, AC, AH, AD and E describe the shift of the function, the parameters BA, BB, BC, BD, BH, FE and F describe the amplitude of the function and the parameters CA, CB, CC, CD, CH, GE and G describe the frequency (Fuhrimann et al. [16]).

3 Machine Learning Algorithms and Image Processing

3.1 Self Organizing Map and Uniform Manifold Approximation and Projection

Self Organizing Map (SOM) is a powerful nonlinear manifold learning method introduced by Kohonen [17]. From a mathematical point of view, SOM transforms (maps) the data from the originally high dimensional space to a low-dimensional space (usually a space of two dimensions), while the topology of the original high dimensional space is preserved. Topology preservation means that if two data points are similar in the high-dimensional space, they are necessarily close in the new low-dimensional space. This low-dimensional space is normally represented by a planar grid with a fixed number of points, which is defined as map. Each node of this map has its own coordinates $(x_{i,1}, x_{i,2})$ and an associated high-dimensional vector $W_i = \{w_{i1}, ..., w_{in}\}$ where the original observed data is represented by n dimensional vectors. In comparison to other data representation methods, SOM has the advantage of delivering two-dimensional maps visualizing smoothly changing patterns of data from the original high dimensional space. This makes SOM a very intuitive method for dimensionality reduction, data clustering and visualization. In addition, SOM can also be used for nonlinear approximation functions (Barreto and Souza [18]).

Uniform Manifold Approximation and Projection (UMAP) is a dimension reduction technique similar to well-known methods such as t-SNE (Van der Maaten and Hinton, [19]) with additional benefit of scalability to large data sets. The algorithm models each high-dimensional object by a two-dimensional point in such a way that similar objects in the high dimensional space are represented by nearby points in the two-dimensional space and dissimilar objects are modeled by distant points. In comparison to SOM, UMAP is not restricted to a fixed number of points in a planar grid and allows representing all input data in an open space, which is beneficial for the representation of clusters. The dimension n of the input vectors of a SOM and UMAP is determined by the clustering objective of the designer. A recent example in the context of structural design was presented by Harding [20] in which three objectives were chosen (twist, height and taper) to represent a solution space of towers with a SOM.

In order to intuitively navigate in a clustered solution space of forms, visually similar forms should be found next to each other. For a single topological setup and only parametric variations, the equilibrium networks can be simply clustered by their nodal coordinates (e.g. for the example of the parametric design space exploration in paragraph 2.2.2, the vector had a length of n=1260). However, when altering the topology, further information has to be included in order to be able to meaningfully separate the clusters. Different clustering methodologies using SOM and UMAP are discussed in section 4.

3.2 Convolutional Auto-Encoders

An auto-encoder is an artificial neural network based on an encoder-decoder model, where the encoder layers first transforms the input into a lower-dimensional representation and the decoder layers reconstruct the initial input. The tuning of this network is generally reached through the minimization of a cost function (In this work the Adam optimization algorithm for deep learning is implemented). The auto-encoder network is trained unsupervised which allows extracting generally useful features from unlabeled data. Auto-encoders and unsupervised learning methods have been widely used in many scientific and industrial applications, mainly solving tasks like network pre-training, feature extraction, dimensionality reduction, and clustering. Deepening the level from only one hidden layer to several layers and using convolutional- instead of fully connected-layers, deep convolutional auto-encoders have been proven to provide well suited results to image processing tasks (Turchenko [21]).

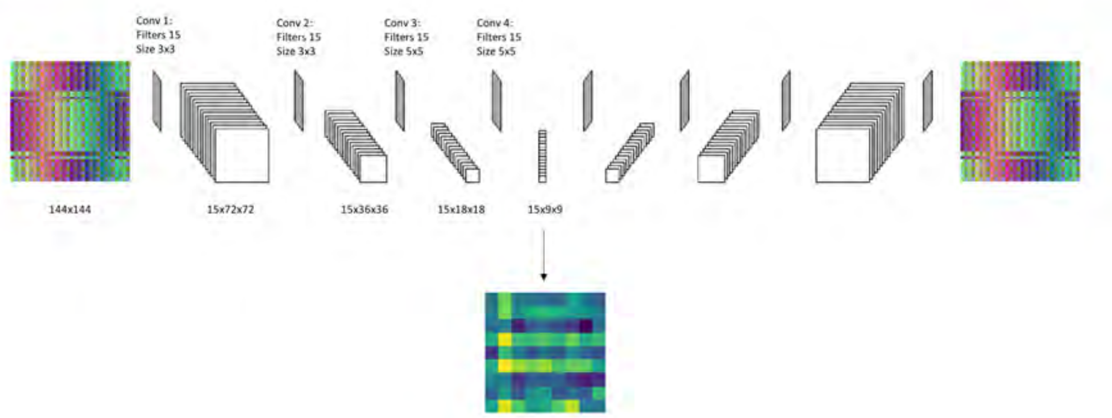

Figure 6: Convolutional Auto-Encoder for Feature Extraction and Dimensionality Reduction.

Figure 6 shows the encoder-decoder setup used in this study. Four convolutional layers are implemented, each consisting of 15 filters, where each filter is representing one convolution (size 3x3 or 5x5). By applying the convolution layers only on every second pixel, each convolution layer is decreasing the size of the image by a factor of two. This results in a final decoded layer of 15 images, each of size 9x9. These images represent the basic components (feature space) of the encoded image in a low dimensional space (from 20736 (144x144) to 1215 (15x9x9)) and can be decoded to the same full-sized image. Next to dimensionality reduction, the decoded layers can be used as features of images. To implement the described encoder-decoder configuration tensorflow [22] is used.

3.3 Generative Adversarial Networks

Introduced by Ian Goodfellow [23], Generative Adversarial Network (GAN) is a class of neural networks implemented by a system of two neural networks competing with each other. The first network is trained to generate images (generator) where that other one evaluates the generated images (discriminator). The discriminator is trained by a known dataset in order to learn which images are the originals and which ones created by the generator. The generator is seeded with a randomized input (e.g. multi-variant normal distribution) and then trained by presenting the generated output to the discriminator. During the learning process, the generator produces better images, while the discriminator becomes more skilled in finding generated images. Generative Adversarial Networks are highly successful for image to image translation tasks. Figure 7 shows a GAN implementation by Isola et al. [24] where a translation from aerial images to a map (left side in Figure 7) is trained to the model. Based on Isola et al., an implementation by Hesse [25] is applied in

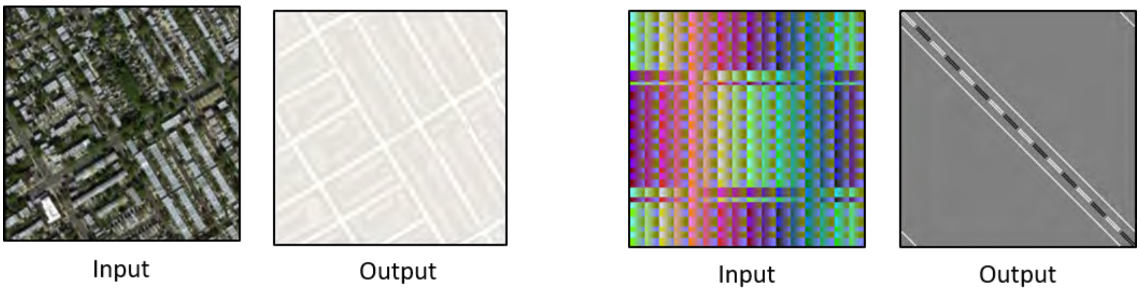

Figure 7: Generative Adversarial Networks, Aerial to Map (left) and Form to Topology (right).

this study to find image to image translations from relative coordinate to topological images (right side of Figure 7).

3.4 Fourier Transformation

The Fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the Fourier or frequency domain, while the input image is the spatial domain equivalent. In the Fourier domain image, each point represents a particular frequency contained in the spatial domain image. The Fourier Transform is used in a wide range of applications, such as image analysis, image filtering, image reconstruction and image compression. Furthermore, image representations in the frequency domain can be used as variables for the machine learning algorithms e.g. to perform classification. However, these classifications come with limitations e.g. scale variations lead to different transformations but translation variations not. Figure 8 shows the Fourier transformations of a scaled white dot in an image (right side) and a transposed point (to the left). The figure shows that the translation of a similar point looks similar in the Fourier domain, scaling the point changes the transformation image.

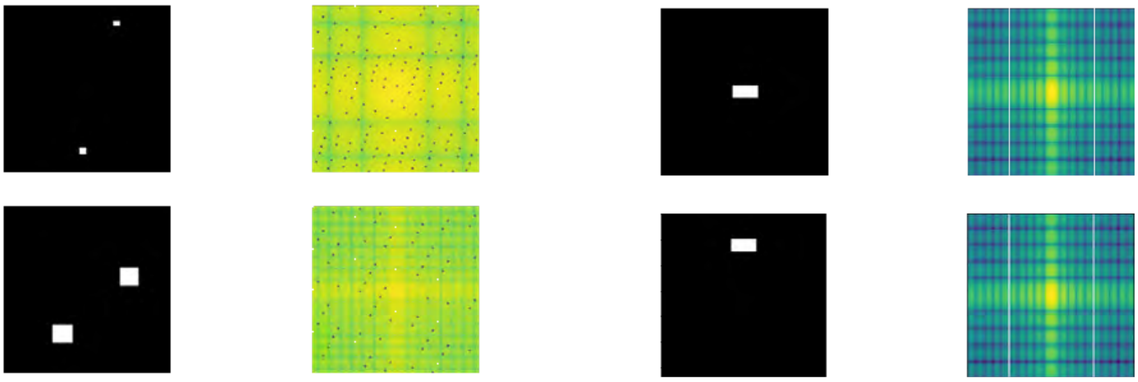

Figure 8: Scale Variant (left) and Translation Invariant behavior of Fourier Transformed Image (right).

4 Data Driven Design, Machine-learning enhanced Exploration

Previous chapters have described a methodology for the automatic generation of solution spaces that inherit diverse equilibrium networks and introduced different machine-learning algorithms. This section discusses methods that bridge this two aspects. The aim is to enable the designers to intuitively navigate within an organized design space and to be able to interact with the solution space.

Therefore, different concepts for the machine interpretation of the generated equilibrium systems are developed and discussed. Based on these interpretations various combinations of the aforementioned machine-learning algorithms are considered.

The overall aim of an organized design space is to allow the designer to intuitively understand the underlying patterns of the clusters. Within a cluster, the structures that are assigned positions close to each other should show comparatively low variation in their properties and a structure further apart should have more divergent properties. The variation of these properties can be calculated to obtain a quantitative study on the efficiency of different clustering methods. However, within the scope of this work, the different approaches are compared qualitatively by the human designer/the author. Three different qualities are taken into account. On the first level, the map has to show a clear underlying pattern. On the second level, the forms close to each other should show a low variation of their visual properties and on the third level, the forms further apart from each other should appear divergent. On top of the description of these design spaces, methodologies to change the generation of spatial equilibrium systems are introduced.

4.1 Data Representation

There are many ways for the representation of three dimensional equilibrium systems. The systems could be simplified with only a few parameters e.g. the total height, width and length of the members. These simplified objectives are fast and easy to understand, however do neglect substantial properties of the networks. This research focuses on a more complex approach of comparing networks as graphs. This allows to conveniently and more precisely compare informations about topology and geometry. In addition, these informations can be directly exchanged with the generating system.

On its most basic level, the spatial network can be described as a set of nodes in space. Additionally including information about the connectivities of the nodes, the network can be represented as a connected graph, its adjacency matrix or its adjacency list. Furthermore, information about the combinatorial state of the members and their metric values can be assigned to the graph which is increasing its complexity.

This work focuses on the clustering of graphs represented in adjacent matrix form. The development of machine-learning techniques has mainly focused on image analysis in the past years. Representing data with images allows to visually understand data, can further be considered as an efficient storage format and allows to make use of existing tools. Therefore, the adjacency matrices are represented in image form. An alternative solution for graph representations, the visual graph description of the spatial network is discussed in the following section and compared to the implemented adjacency matrix representation.

4.1.1 Adjacency Matrix Representation

The adjacency matrix of a graph is an abstract way to represent and compare information of equilibrium networks. This matrix (Figure 9) describes the connectivity of nodes (binary values). When increasing the range of values from binary to real, another resolution of information (e.g. metric values of trail lengths) can be included into the matrix. The dimensionality of this matrix can be scaled up to infinite dimensions and therefore allows to embed the coordinates of the nodes and or the metrics of the external forces.

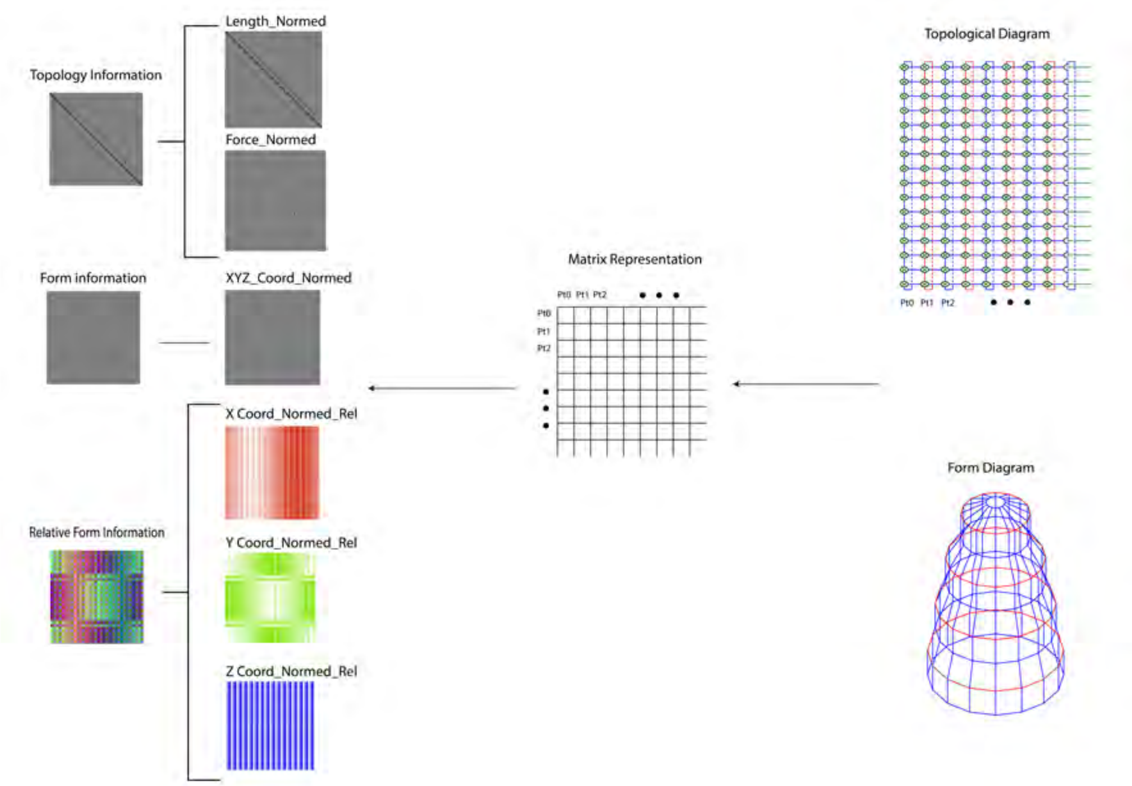

Figure 9: Image Representation (left) of the Adjacency Matrix (middle) of a Tower (right).

Figure 9 shows different possibilities of image representations (left) for the adjacency matrix of a pre-defined topological diagram and a corresponding form diagram (right). Combining the information of nodal connectivity with the metric values of trail and deviation members, a **topological image representation** is created (top). The nodal coordinates in the form diagram (3-dimensions x,y,z) can either be represented with their actual position (middle, **form information**) or relative to each other (bottom, **relative form information**). Each of the coordinate dimensions is represented with a color code: red, green or blue. The informations of each image are normed according to equation 1.

$$x_{i,j,normed} = \frac{x_{i,j} - minval}{maxval - minval} \tag{1}$$

The **topological images** are normed respectively to their trail or deviation correspondence:

$$minval_{trail,dev} = -max(|min_{i,j}(X[i][j])|, max_{i,j}(X[i][j])) \tag{2}$$

$$maxval_{trail,dev} = max(|min_{i,j}(X[i][j])|, max_{i,j}(X[i][j])) \tag{3}$$

The **form images** are normed respectively to the coordinate dimension x,y,z:

$$minval_{x,y,z} = -max(|min_{i,j}(X[i][j][k]|, max_{i,j}(X[i][j][k])) \tag{4}$$

$$maxval_{x,y,z} = max(|min_{i,j}(X[i][j][k])|, max_{i,j}(X[i][j][k])) \tag{5}$$

The **relative form images** are normed separately in each row i, respectively to the coordinate dimension x,y,z:

$$minval_{i(x,y,z)} = -max(|min_{j}(X[i][j][k]|, max_{j}(X[i][j][k])) \tag{6}$$

$$maxval_{i(x,y,z)} = max(|min_{j}(X[i][j][k])|, max_{j}(X[i][j][k])) \tag{7}$$

To compare coordinates of two equilibrium networks with different topologies (altering dimensions of information), a node searching algorithm is implemented, which is increasing the dimension of every structure to the highest occurring dimension. The algorithm is replacing the missing coordinates with the closest neighbor in the same layer. If there is no node existing within the same layer, the next node in the trail is considered. This supports the comparison of the low dimensional with the higher dimensional graphs without having to change their general resemblance. The resulting images for topology, form and relative form are then tested with different clustering methodologies.

4.1.2 Visual Graph Detection

As an alternative to image representations of the adjacency matrix, the graphs can be also described using either 2D pixel or 3D voxel representations. Structures with 2D and 3D image representations can be analyzed and then clustered by detecting their edges (e.g. using the canny edge detector by Open CV). Working with 2D representations, spatial information is simplified and can be lost (e.g. rotational symmetry). 3D representations of the visual graphs would therefore be more ideal to represent the structures. Since edge detection itself is not able to differ between the combinatorial states of members, a second algorithm needs to be considered to extract this feature. 3D voxel representations require higher storage capacity compared to the adjacency matrix image representations. Further, it is less convenient to include information of parametric properties such as the external forces into the formulation. Therefore, this method is not implemented in the scope of this work, would however be interesting to investigate and compare to the chosen adjacent matrix image representation in future work.

4.2 Image Processing for Form and Topology Images

Two image processing methodologies introduced in section 3 are applied to process **topological** and **relative form images**. The clustering quality of the processed images is examined and later compared to the original image representations (the comparison can be found in Table 4).

4.2.1 Fourier Transformation

Topology and **relative form images** are decomposed into its sine and cosine components using Fourier transformation (using fft by Open CV [26]). The zero frequency component is shifted to the middle of the image. The transformed images can be used as an abstraction of the images to eventually find features for clustering and to train neural networks. However, the tests exhibited no clear patterns when

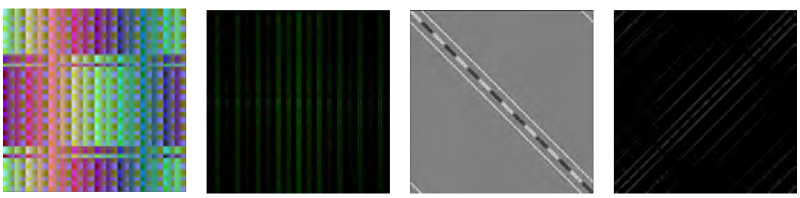

Figure 10: Fourier Transformation of Relative Coordinate Image and Topology Image.

training a Self Organizing Map with these transformed images. To give an example of such an unstructured clustering, Figure 11 shows a family of suggested similar equilibrium structures by the map. Although this test seems not to work, the images increase the learning rate of the encoder-decoder network.

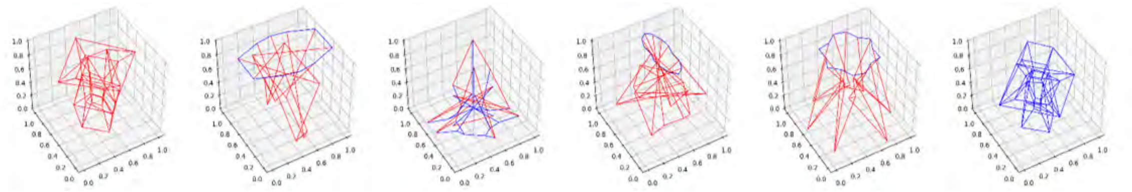

Figure 11: Form Family suggested by a trained SOM using images in the Frequency Domain.

4.2.2 Convolutional Auto-Encoders

To reduce the dimensionality and improve the clustering of **relative form images**, the auto-encoder network described previously is applied for feature extraction in order to train a SOM. Figure 12 shows the input (Top) and output (Bottom) of images representing relative from information. As can be seen, the decoded Images are not ideal representations of the input. Experiments altering the depth, sizes and training epochs of the algorithm have shown that the learning rate is well adjusted for the setup represented in Figure 6. Training a SOM with encoded **relative coordinate images** (Figure 18) results in a improved clustering quality compared to a SOM trained with **form images** when the number of vertices in compared graphs is altering. Figure 13 shows a form family of a trained SOM with encoded

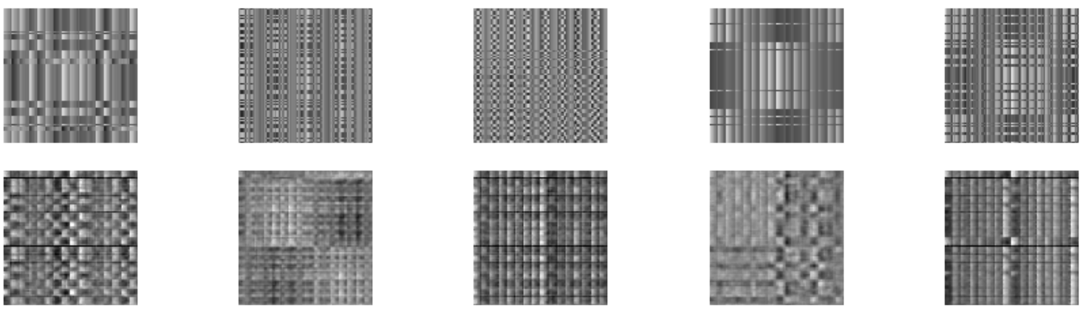

Figure 12: Input (Top) vs. Output Images (Bottom).

relative form images. The equilibrium networks appear to be spatially similar but are topologically different. Training a SOM with encoded **topological images** does not result in readable cluster patterns.

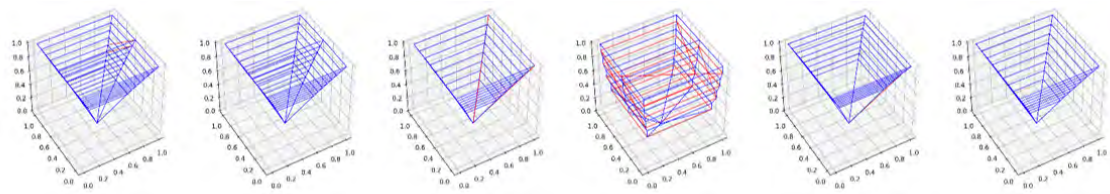

Figure 13: Form Family of a SOM trained with Encoded Relative Form Image representations.

4.3 Clustering Methodologies

The clustering methodologies within this chapter are divided in two parts. The first approach addresses clustering for structures with topological variations while the second one addresses clustering for structures with only parametric variations. An approach, in which both aspects are mixed is possible but not implemented in the present work.

4.3.1 Parametric Clustering

Varying the metrics within a specific topological and interfacial setup, the number of nodes and connections will remain constant. As a straight-forward approach, it is suggested to use solely the geometric information (the **form image** in Figure 9) to cluster and navigate within the solution space. Figure 14 shows a Self Organizing Map trained with data that was generated based on the parametric variation described in chapter 2. Every input structure is presented to the trained map and

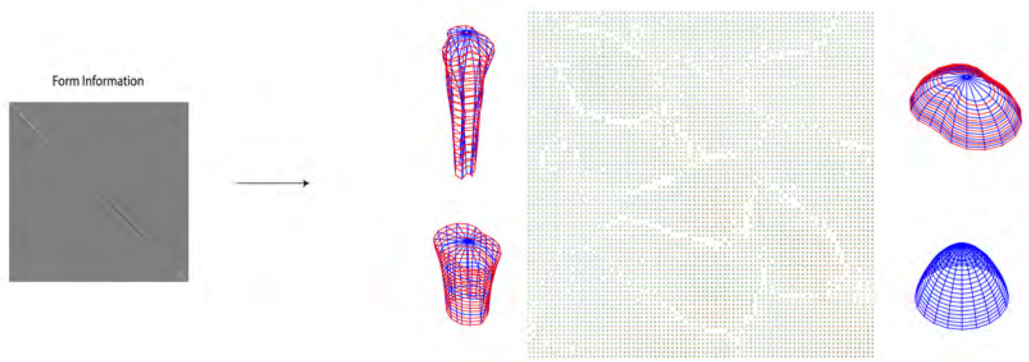

Figure 14: SOM trained with Form Image information.

assigned to its best matching unit (bmu) in the map. The nodes of the map are then represented by one of the assigned structures. All the structures assigned to a single bmu represent a form "family" together. An example of such a family is shown in Figure 15. Within this "family", the networks show a well defined spatial correlation.

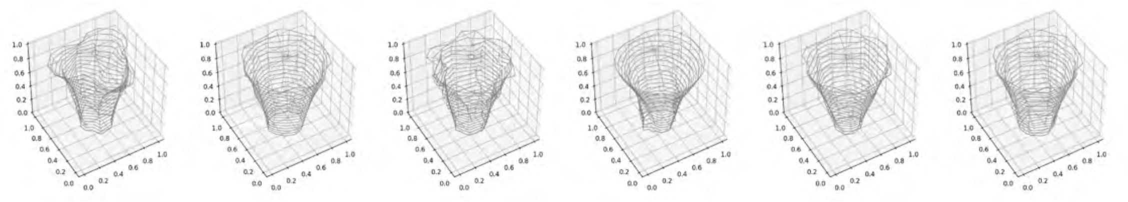

Figure 15: Form Family of a single node in the trained SOM.

4.3.2 Topological Clustering

Comparing networks with topological variations brings difficulties to the clustering process. As discussed, lower dimensional forms are scaled to the dimension of the largest structure by a node searching algorithm within this thesis. However, not only the dimensions of the equilibrium networks differ, but also their combinatorial state and their connectivity.

Image Representation	**Clustering quality**
Topological image	Clear clustering patterns, spatial configurations not included and therefore variations within a form family.
Form image	Clear clustering patterns for constant node number, topological information not included and therefore variations within a form family
Combined Form and Topological image	No clear clustering patterns can be detected
Relative form image	Could not be tested because of computational memory errors
Fourier transformation of Topology image	No clear clustering patterns can be detected
Fourier transformation of Relative Form image	No clear clustering patterns can be detected
Auto-encoder Topological image	No clear clustering patterns can be detected
Auto-encoder Relative Form image	Clear clustering patterns, topological information not included and therefore variations within a form family

Table 4: Table of image representations and the resulting clustering quality.

A qualitative study that combined different image representations was conducted.

The results can be seen in Table 4.

Mixing topological and geometric information within one image apparently leads to unreadable maps without a clear underlying pattern. Therefore, a hierarchical approach is suggested, where a first cluster is trained with **topological images** of the equilibrium networks. Based on a selected region the algorithm is then guided in a second step by either **form** or **relative form images**. Figure 16 shows a self organizing map trained first with **topological images** only. In the map, the difference between the combinatorial states and the density of connections can clearly be distinguished from only tension (top left) to only compression (bottom right). This allows the designer to decide on the overall structural system he would like to consider e.g. structure dominated by tension with only a few connections, before investigating the spatial geometric potential of these structures.

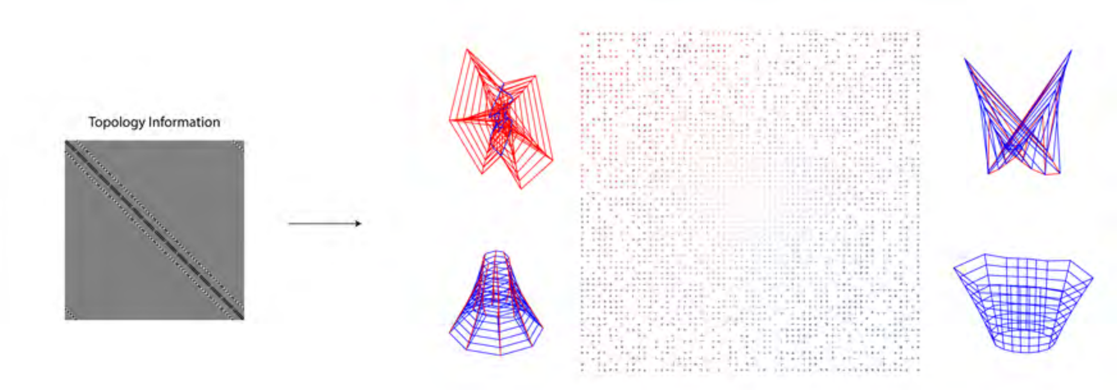

Figure 16: SOM trained with Topological Image.

As described in the previous section, the nodes of a SOM can be regarded as the representatives of form families. A family of a single node in the trained SOM is shown in Figure 17. Hence, the equilibrium networks within this family indicate a well defined topological similarity. Since no metric information is included in the input image, a spatial variety between the structures within the family apparently exists.

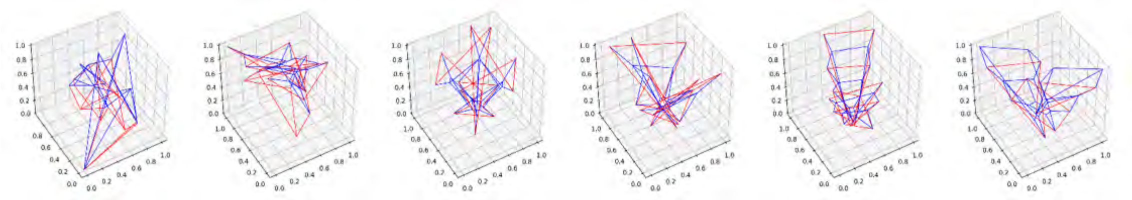

Figure 17: Form Family of a single node in the trained SOM

After selecting a region of topological similarity, the equilibrium networks can then be clustered by informing the map with the **relative position** of the latter. This allows the designer then to distinguish between the spatial resemblance of forms. Figure 18 shows a trained SOM with encoded images representing the **relative form information**.

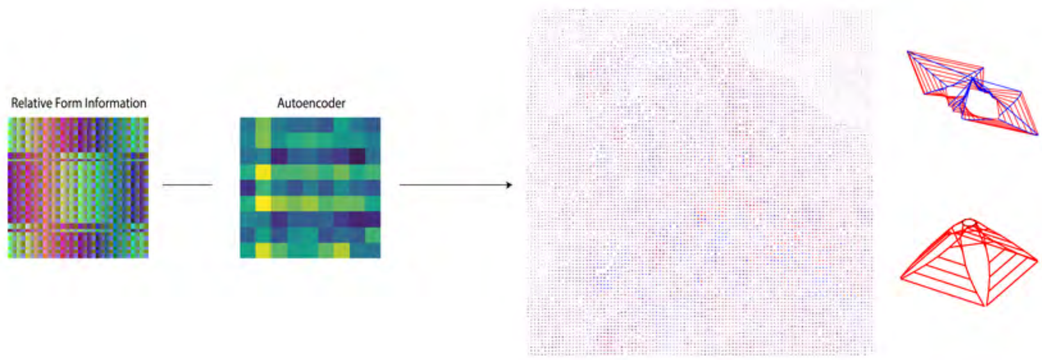

Figure 18: SOM trained with Relative Form Image.

The equilibrium networks in the form family shown in Figure 19 are ultimately resembling each other in their topological and their spatial configuration. Using the proposed hierarchical clustering procedure, a well defined solution space of equilibrated structures can be obtained (presented to the designer).

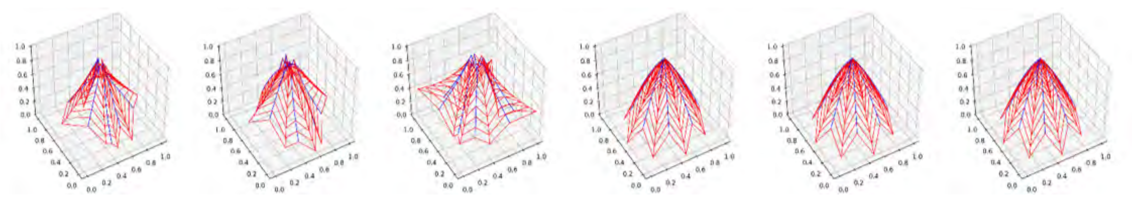

Figure 19: Form Family of a single node in the trained SOM.

4.4 Data Generation Methodologies

As shown in the previous sections, machine-learning algorithms can be applied to cluster a solution space. Additionally, these algorithms can be used to gain insights between the input and the output of complex systems and even to create new data (generative models).

4.4.1 Parametric Informed Data Generation

Using a well-defined set of input parameters for the parametric exploration, an analysis of the impact of these parameters on the diversity of the solution space can be conducted. Therefore, a UMAP algorithm is trained with the data generated from the setup described in section 2.2.2. Each point of the scatter plot in Figure 20 (left) corresponds to a single input form, which is modeled as a two-dimensional point by the algorithm. The point clouds in Figure 20 indicate similar forms in the Euclidean space and the distance between the clouds shows the metric resemblance between the clusters. Hence, nine different clusters can clearly be distinguished.

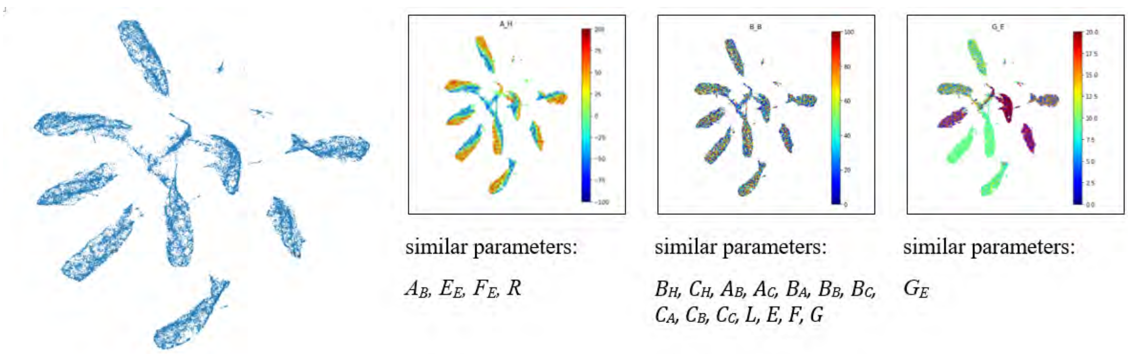

Figure 20: Clustered space of UMAP (left) and UMAP Parameter Representation of Three Different Classes-

To understand the correlation between the CEM input parameters and the generated equilibrium forms, a visualization of the different parameters for the trained UMAP is carried out. Three qualitatively different classes of correlation can be detected, as shown on the right side of Figure 20. (Local) Parameters that direct the resulting equilibrium form within a cluster into a specific region (left), parameters that are arbitrary distributed within each cluster (middle) and (Global) parameters that clearly describe to which cluster an equilibrium form belongs (right). Based on these insights, data can be generated more precise. In the present case, by varying only the parameters that increase the size of a selected cluster or increase the diversity of the entire space.

4.4.2 Generative Models

The idea of a generative model is to learn the most probable output for a given input. This can be useful for data generation e.g. when aiming to create new equilibrium systems without using the CEM algorithm itself but by training a neural network. Variational convolutional auto-encoders are an example of such a method. These networks are similar to the introduced auto-encoder networks but a constraint has to be added to the encoding network. This constraint forces the encoder to generate a latent (most inner) representation that roughly follows a unit Gaussian distribution. This behavior can be used to generate new equilibrium networks by sampling a representation from the unit Gaussian and pass it into the decoder. Variational auto-encoders are not implemented in the scope of this research, however could be used in future research in order to create new (and-or in-between) structures within a learned space.

Another use of generative models is the translation of coordinate information into topological information. This translation can help designers to understand and alter the load bearing behavior of an existing building or a planned geometry. This can be of interest when searching ideas for a specified design brief. In this case the infinite solution space of clustered maps can not provide direct solutions. A translation could therefore be used to generate spatial networks within the geometric boundaries of the task.

Defining spatial nodes of a basic geometry, a **relative form image** is generated (middle). A neural network can then be trained to translate from this geometrical information to a desired topological configuration and its metric values (**topological image**, right). Finding this translation, an equilibrium network can be found which could in a next step either be translated into a real world application or used to understand and alter the load bearing behavior of an existing structure (e.g. the Taipei tower in Figure 21 (left))

This translation network is implemented as an adaption of the auto-encoding net-

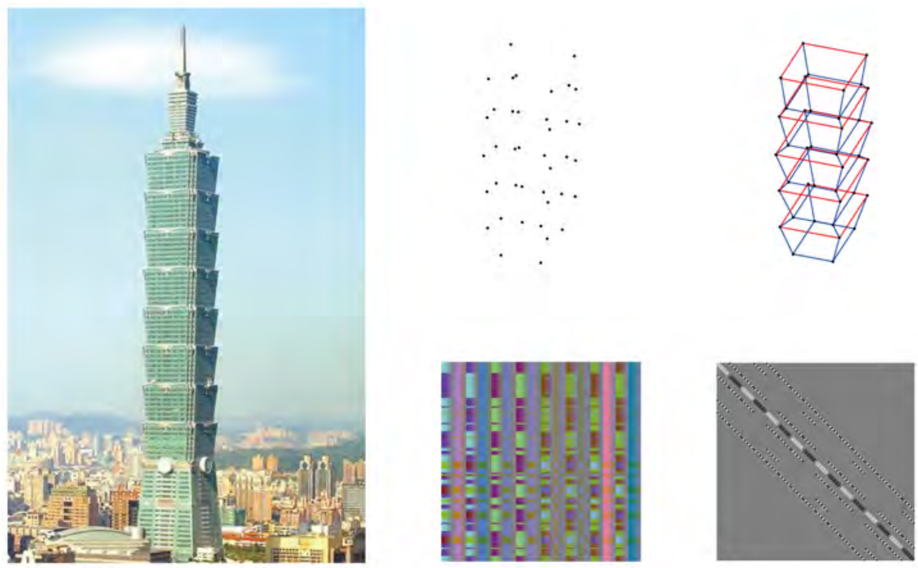

Figure 21: Example: From Spatial Nodes to Topology and the Taipei Tower.

work. Therefore, the encoder-decoder system is presented **relative coordinate images** as an input and trained against **topological images** as an output of the decoding layer. The optimizer is training the network to describe the translation between the two images. The trained network can for example be used to translate the **relative coordinate image** of the Taipei tower shown in Figure 22 (middle, left). The translation results in a uniform gray scale image which represents a zero (no connection) configuration. Transforming the images using the Fourier transform (middle, right), a transformation pattern can be learned by the network. However, converting the images back to its spatial domain, the translational invariant behavior of the conversion prevents a distinct description of the suggested topology. A second, more promising approach using GAN is implemented and tested. This algorithm is able to learn a translation between the two images. The translation from the Taipei **relative coordinate image** to a **topological image** is shown in Figure 22 (right). However, the image is describing a topological configuration which is not readable by CEM or comparable to the desired configuration shown in Figure 21, right. In conclusion, a more accurate translation is necessary to obtain a topological definition of a desired geometry.

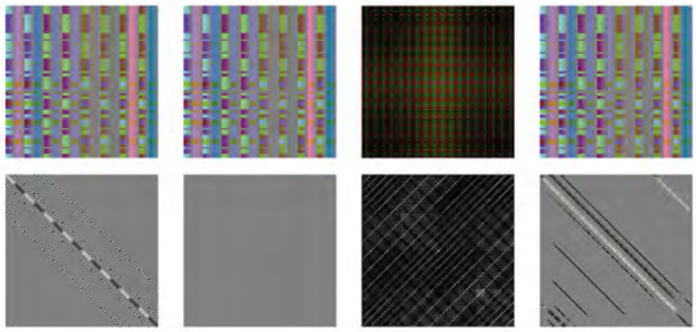

Figure 22: Relative Coordinate Image as Input (Top), and Topological Image as Output (Bottom) trained with a Convolutional Neural Network (middle) and a Generative Adversarial Network (right).

5 Interactive Form Explorations and Design Procedures

Within the scope of this thesis, 100GB of data resulting in approximately 2 million equilibrium networks have been generated using 30 different CEM configurations for the topological exploration.

In order to study these systems, an interactive procedure for the suggested hierarchical clustering approach is implemented and exemplary represented in Figure 23. In a first step, a Self Organizing Map is reducing the size of each of the 30 configurations to a size of approximately 400MB. By this the 2 million equilibrium networks can be reduced to 200'000 representatives. In a next step, the spatial structures are clustered and visualized by their **topological images**. An interactive scatter plot is implemented, which allows to select the desired topological configurations. All the selected structures are then clustered using their **relative form coordinates** in order to correlate spatially similar structures. For the sake of clear distinctions between the different clusters Umap is used. In a last step, the python library IPyvolume is used to render an interactive 3 Dimensional map, in which every node is visualized by the representative equilibrium network.

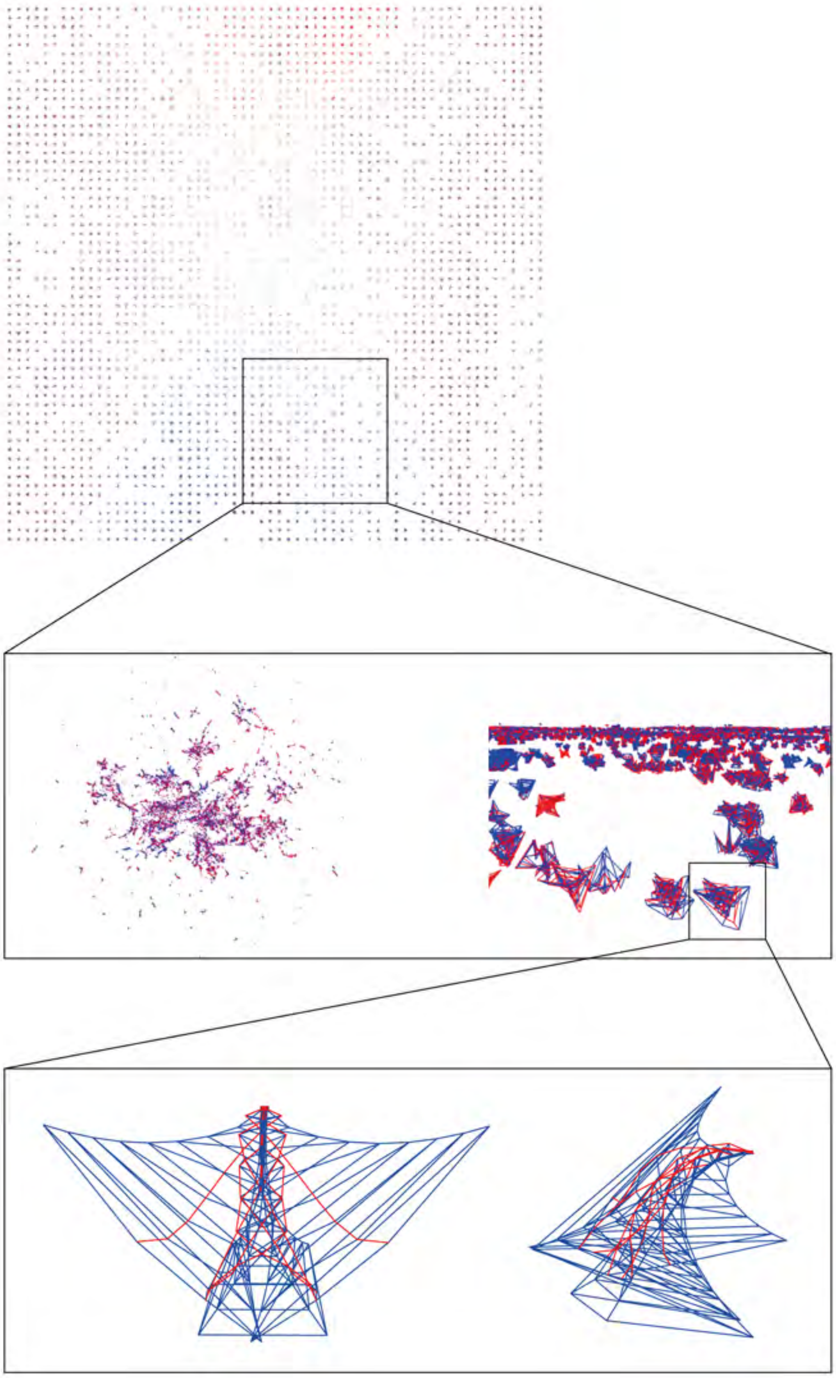

Figure 23: Example of Hierarchical exploration using SOM (topological and metric infomration), UMAP (node information) and Ipyvolume (3D visualization).

The unpredictable and intriguing shape of the network that is exemplary chosen, shows that this combination of methods can lead to inspiring and unforeseen structural solutions that go beyond any typological border. A physical sculpture that was inspired by this form can be seen in Figure 24. Next to using the geometric information of the reference design, the sculpture was materialized using the information of the load bearing behavior (sustaining vertical loads) of the reference design. Therefore, the compression members of the sculpture are built as a steel frame which is pulled back by a tension chord in order to achieve that the steel frame is able to stand on two single supports.

Figure 24: Sculpture of the "Bird" Structure.

6 Conclusions and Outlook

Data-driven techniques bring a new perspective to the field of structural design, shifting the focus from the design of a singular artifact to the design of a system that inherits many artifacts. The architecture of the data generating system itself is becoming the limiting parameter of this approach. The main challenge for these generating systems is on the one hand to control the generation in order not to have too many unusable data, but on the other hand to not to restrict the system too much in order to get innovative solutions. Hence, a balance between the two aspects needs to be found. A balanced system that is able to generate diverse and meaningful artifacts.

This work focuses on Combinatorial Equilibrium Modeling as the generating tool, which allows to create equilibrium networks for the early design phase. Understanding diverse reference systems such as towers, bridges, stadiums and shells, a basic composition is implemented, which enables to find these structures within the solution space. This basic representation is then altered on two conceptual levels (parametric and topological). These alterations are allowing the solution space to grow in a controlled manner without restricting it to a specific typology, but creating new and unexpected equilibrium systems. The structural designs generated with this methodology appear however mainly unorganized and very complex and therefore "unreadable" for a human designer. This leaves the question if a more appropriate tool or methodology exists for the same task.

To make all the solutions easily accessible and comprehensible to the designer, the big challenge is to find an organized and well structured representation. The difficulty of this task is to define a non-trivial clustering objective for non labeled data. Within this thesis, a qualitative study of different methodologies based on various data representations and clustering techniques are performed. To compare the different methodologies, a criterion was defined. This criteria regards methodologies as "good" when they lead to a design space representation which follows a clear pattern and the visual distance of the structures in the design space describes the variation of this pattern. As a solution, a hierarchical approach is proposed. This approach starts with **topological image** clustering, and continues with clustering of **relative coordinate images**. This allows a designer to get a good overview of a complex and large data space and eventually supports him in the identification of unicorn solutions.

The equilibrium networks generated and organized by the proposed methodology are exemplary used to find inspiring design ideas. Within the enormous amount of useless/unreadable systems, the "Bird" could be singled out and identified as such an idea. This "Bird" is materialized as a sculpture by including its structural information, closing the circle of designing with data.

The question remains, whether such a data-driven approach is able to replace the human designer as such and to what degree it can be used in a real-design scenario?

"At our current level of technology, it appears that the best defense against automation is creativity." (VentureBeat [27])

The research presents an approach to overcome standard design methodologies and their limitations such as physical models. It could be proved that it is possible to find new design ideas with this method. Whether such an approach is helping designers to consider various different systems for a single task is still questionable. The abstract formulation of a meaningful but still surprising generating system seems

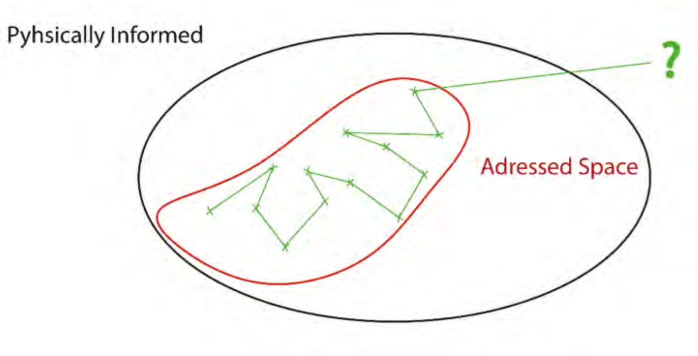

Figure 25: Pattern between the Unicorn Solutions.

very difficult. Figure 25 shows the unicorn solutions as green crosses in the physically informed space. To be able to compete with the human as an intelligent and intuitive designer, the machine needs to learn the relations between these solutions. This concludes in a new research question.

"Can we train a machine to find a relation of unicorn solutions and become more efficient than the intuitive designer?"

References

[1] Nigel Cross. *Designerly ways of knowing*. Springer, 2006.

[2] Jesse Louis-Rosenberg and Jessica Rosenkrantz. Nervous system, https://n-e-r-v-o-u-s.com/. 2018.

[3] Benoit B Mandelbrot. *The fractal geometry of nature*, volume 173. WH freeman New York, 1983.

[4] George Stiny. Introduction to shape and shape grammars. *Environment and planning B: planning and design*, 7(3):343–351, 1980.

[5] Heinz Isler. Concrete shells derived from experimental shapes. *Structural engineering international*, 4(3):142–147, 1994.

[6] Frei Otto, Rudolf Trostel, and Friedrich Karl Schleyer. *Tensile structures; design, structure, and calculati on of buildings of cables, nets, and membranes*. The MIT Press, 1973.

[7] Caitlin Mueller. Designing with data: Moving beyond the design space catalog.

[8] Caitlin Mueller and John Ochsendorf. An integrated computational approach for creative conceptual structural design. In *Proceedings of the international association for shell and spatial structures (IASS) symposium*. IASS Wroclaw, Poland, 2013.

[9] Kristina Shea and Jonathan Cagan. The design of novel roof trusses with shape annealing: assessing the ability of a computational method in aiding structural designers with varying design intent. *Design Studies*, 20(1):3–23, 1999.

[10] Juney Lee. Grammatical design with graphic statics: Rule-based generation of diverse equilibrium structures. *Master Thesis*, 2015.

[11] Patrick Ole Ohlbrock, Pierluigi D'ACUNTO, Jean-Philippe JASIENSKI, and Corentin FIVET. Constraint-driven design with combinatorial equilibrium modelling. *IASS 2017*, 2017.

[12] Patrick Ole Ohlbrock and Joseph Schwartz. Combinatorial equilibrium modelling. In *Proceedings of the IASS symposium*, 2015.

[13] T Wortmann. *Representing Shapes as Graphs*. PhD thesis, Massachusetts Institute of Technology, 2013.

[14] Davide Tanadini. Topological explorations in high-rise design. *ETH Zuerich 2017*, 2017.

[15] Andrea Biancardi. Digital equilibrium design for curved bridges. *ETH Zuerich 2017*, 2017.

[16] Fuhrimann Lukas, Moosavi Vahid, Ohlbrock Patrick Ole, and D'Acunto Pierluigi. Data-driven design: Exploring new structural forms using machine-learning and graphic statics. 2018.

[17] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982.

[18] Guilherme A Barreto and Luís Gustavo M Souza. Adaptive filtering with the self-organizing map: A performance comparison. *Neural Networks*, 19(6-7):785–798, 2006.

[19] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[20] John Harding. Dimensionality reduction for parametric design exploration. *Advances in Architectural Geometry 2016*, pages 274–87, 2016.

[21] Volodymyr Turchenko, Eric Chalmers, and Artur Luczak. A deep convolutional auto-encoder with pooling-unpooling layers in caffe. *arXiv preprint arXiv:1701.04949*, 2017.

[22] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016.

[25] Christopher Hesse. https://github.com/affinelayer/pix2pix-tensorflow. 2017,accessed: 16.06.2018.

[26] Itseez. Open source computer vision library. https://github.com/itseez/opencv, 2015.

[27] Anthony Wood. Will ai replace creative professionals? 2017.

List of Figures

List of Tables