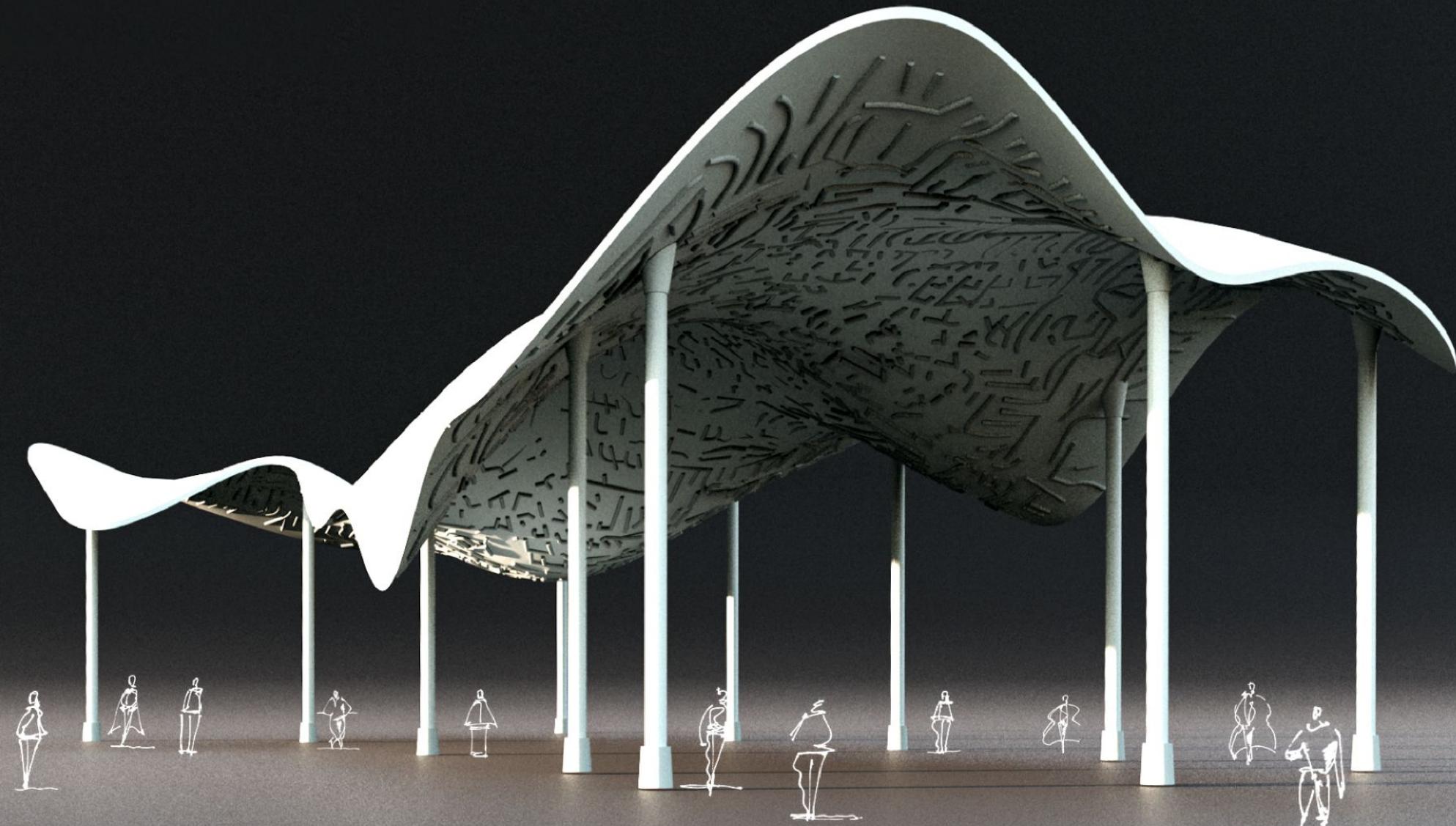


Computing in Architecture

Tzu-Ching Wen | Pinaki Mohanty

Harmonic Sculpting of Concrete Shell



I. INTRODUCTION

- GOAL
- BACKGROUND
- PROBLEM SET-UP



GOAL

To find most optimized shape(s) for free-form concrete shell using novel strategies.

Shape Optimisation using NURBS

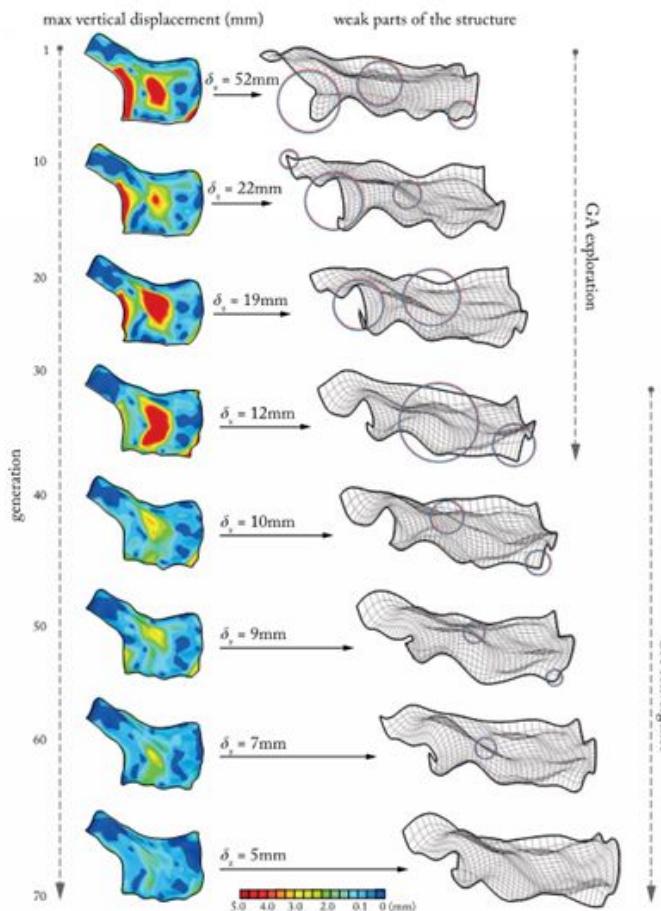


Fig 1. Overview of the morphogenetic design process of the Kakamigahara Crematorium using evolutionary algorithm [3]

The shell forms found using physical models tend to be rather unvarying for a fixed plan and support conditions. But with the development of more robust optimization algorithms the field of generative design has become fairly broad. One such example where it has been successfully utilised and designed is the Kakamigahara Crematorium in Gifu, Japan [3]. The free-form surface is represented using NURBS with sufficient number of control points to control the shape. The advantage of using NURBS discretization over mesh is that the parametric variables for optimization is greatly reduced and adequate degree of continuity is maintained too.



Fig.2: Kakamigahara crematorium, Gifu, Japan, designed by Toyo Ito together with Mutsuro Sasaki. [3]

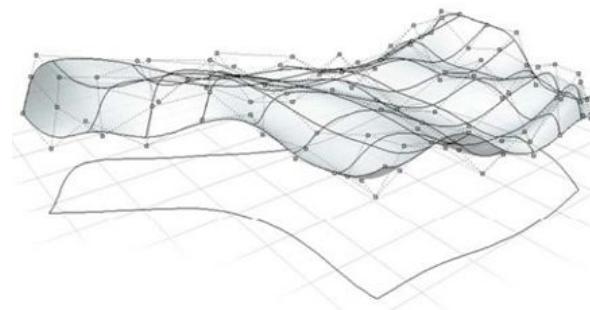


Fig.3: NURBS representation of the crematorium roof. [3]

Eigen Shells

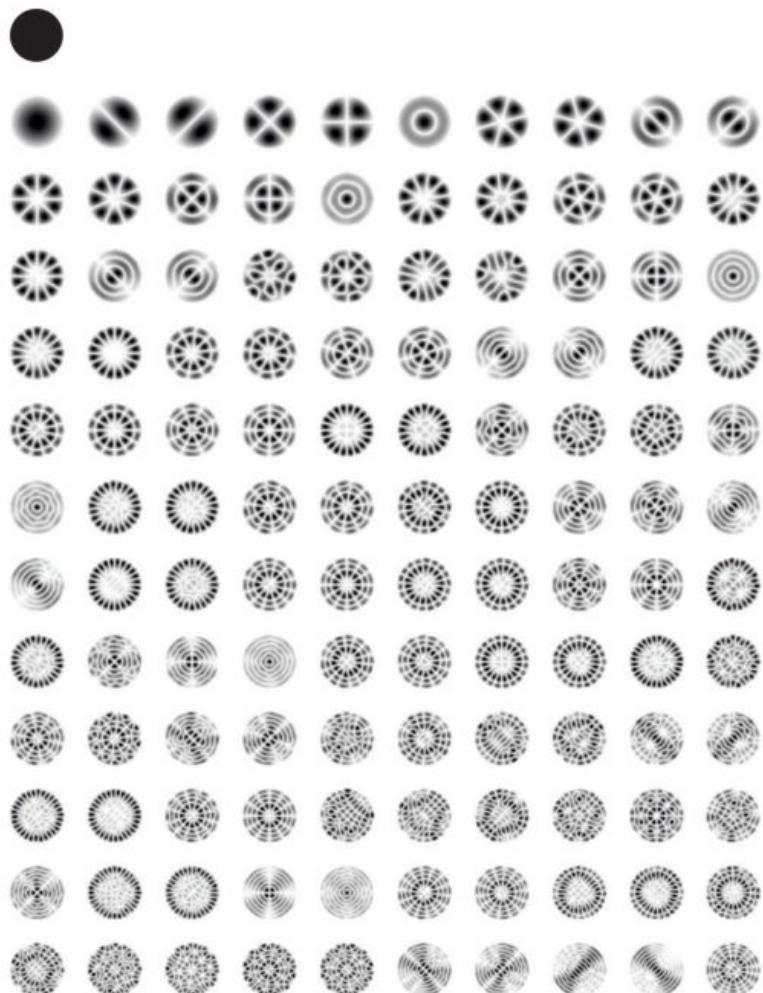


Fig.4: First few eigenfunctions of a circular potential. [4]

Another way of choosing the parametric variables for shape optimisation is to find a set of shapes that are intrinsically linked to geometry of the shell so that the designer can choose from these set of shapes or a combination of them which are visually satisfactory. For this purpose one could use eigen functions of a discrete lapalcian. Each of these eigen functions represents an eigen shape giving a displacement variable and a linear combination of these shapes using weight factors controls the shape of the shell structure. So clearly the weight factors become the parameters for optimization.

Although grasshopper plugins like Millipede allows one to evaluate eigen vectors, however it has no features to include the boundary conditions. The plugin is virtually a black box and extracting the matrix to manipulate it seems impossible. We therefore then chose to utilise the modal shapes which not only takes into account the mass and area properties of the shell but also quite accurately deals with different boundary conditions. The suitability of this approach was well studied and documented by Cecilie Brandt-Olsen in her MPhil thesis at the University of Bath. [5]

Eigen Shells

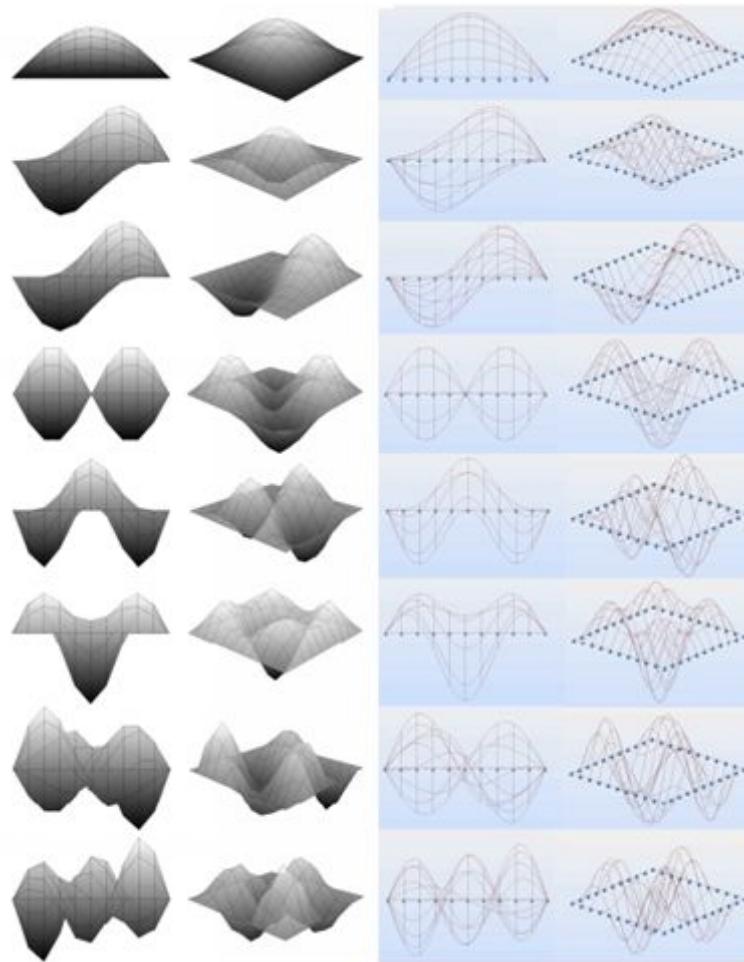


Fig. 5. Comparison between the first eight mode shapes of a flat square mesh calculated from an eigen decomposition of the graph Laplacian (left) and a modal analysis in Autodesk Robot (right) [5]

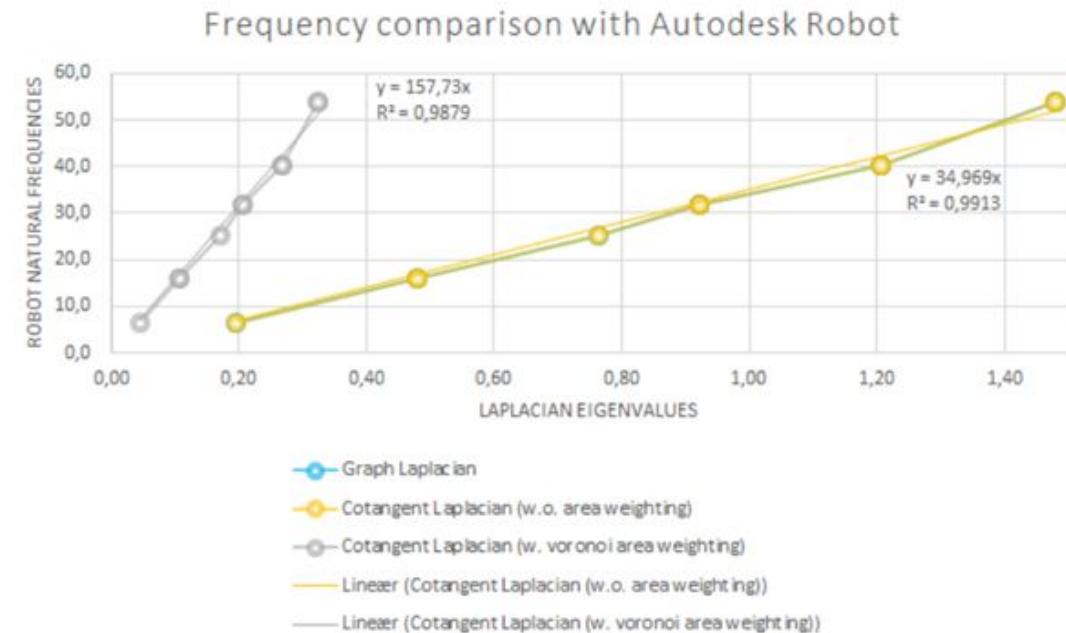


Fig 6. Frequency comparison between the results obtained from a modal analysis in Autodesk Robot and the eigenvalues from an eigen decomposition of the different discretisations of the Laplacian matrix for the ten first modes of a flat square mesh. A linear relation is observed. [5]

The relationship between the modal frequencies obtained using both the methods is observed to be linear which establishes the fact that the weight factors can still be used as parameters of optimization. We therefore used the Karamba plugin to evaluate the mode shapes and proceed.

Problem Set-Up



Fig. 7. Google maps satellite view of the site.



Fig. 9. Google Street view of the site.

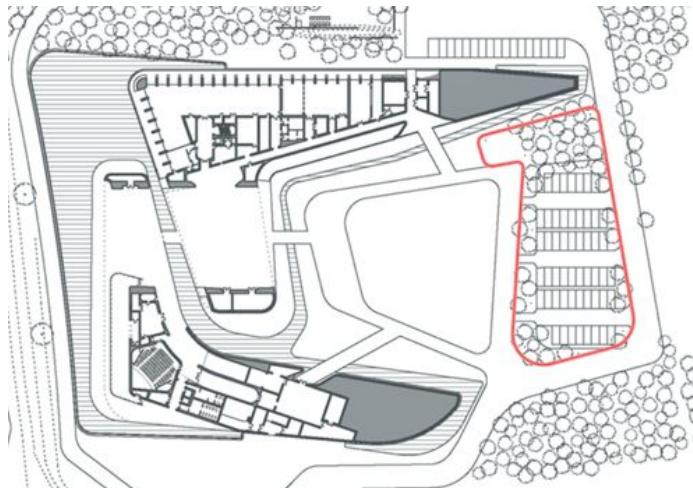


Fig. 8. Site drawing. The exact location is marked in red [7].

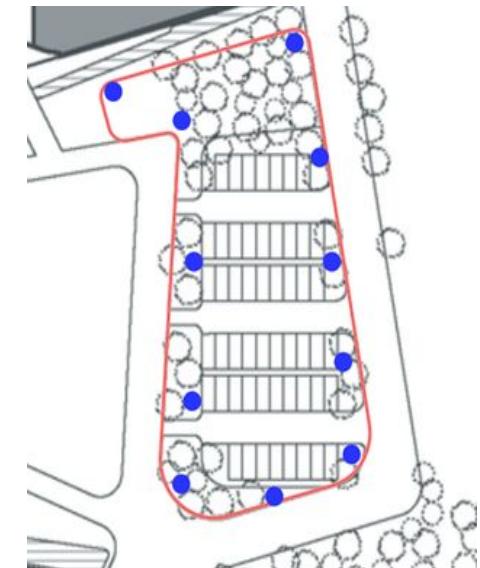


Fig. 10. Column positions of the proposed shell structure [7].

For our assignment we chose to select a parking garage site near the famous Sun-Moon Lake in Taiwan in order to test our shell design method. Figure 7 and Figure 8 show the site location and other aspects clearly. We wish to create a shed made of concrete shell over the parking lot as marked red in Figure 6. This serves as the starting flat surface for computing the modal shapes. The columns which act as point supports for the shell structure are appropriately placed to give least hindrance to the parking lot with the highest span of around 30m.

Workflow

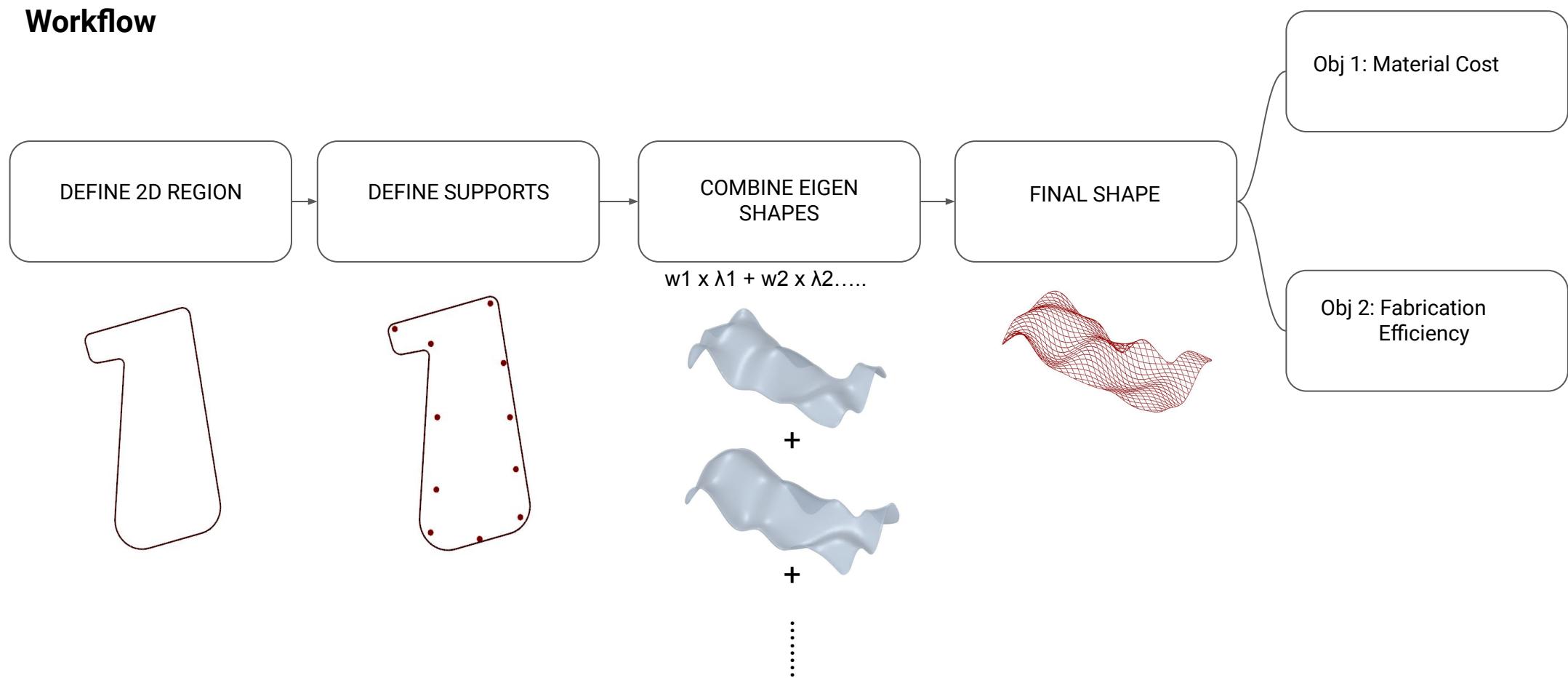


Fig. 12. Workflow of the optimization process.

As shown in the figure above the basic workflow is as follows. We first define the 2D region or the flat shape on which the modal analysis is to be carried out. After defining the boundary conditions or support locations modal eigen shapes are evaluated using Karamba. The eigen shapes linearly combined using suitable weight factors. These weight factors serve as the parameters for optimisation. The final shape is then evaluated to get the material cost and the fabrication efficiency, which will be the first and the second objective respectively.

Parameters

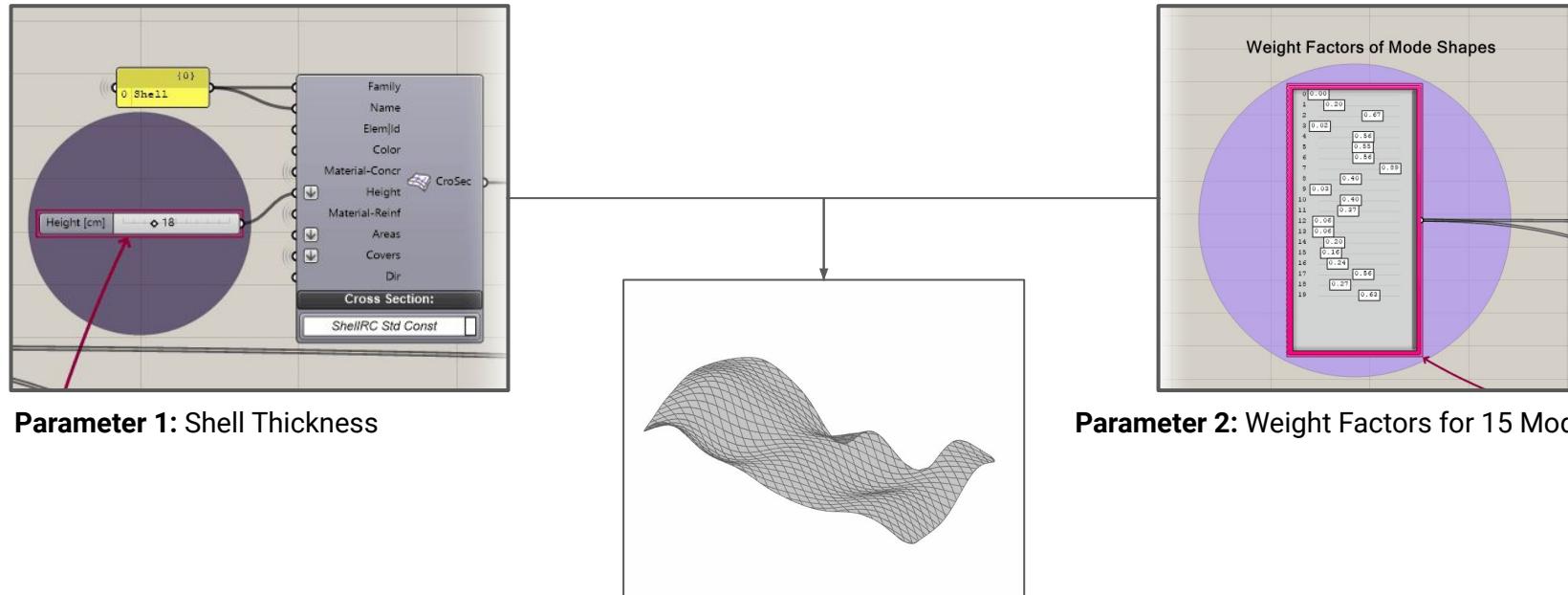


Fig. 13. Parameters controlling the shape of the shell structure.

The shell thickness serves as the first parameter and since we chose the first 15 modes, the weight factors for each mode shapes become a parameter. Therefore in all there are 16 parameters.

Objective 1: Material Cost

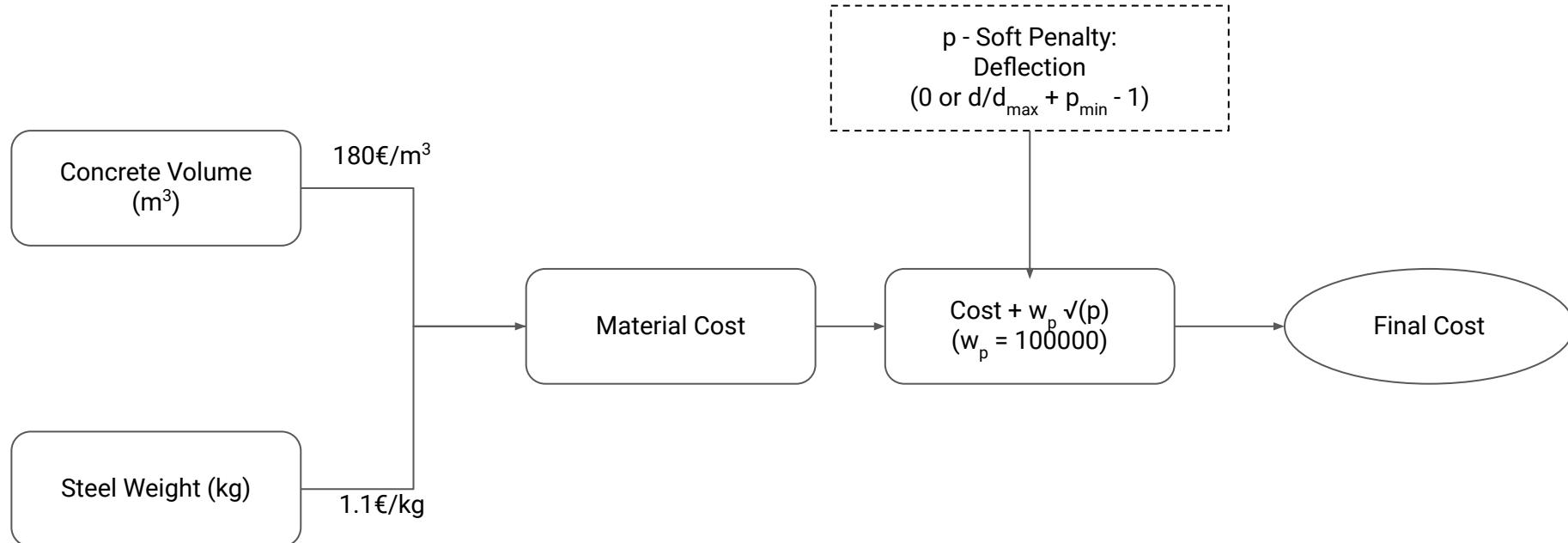


Fig. 14. Summary of the evaluation procedure for material cost.

Material cost includes the concrete volume and the steel weight. The rates used to compute the total cost are taken from the *Cost Construction Analysis* report by Bilfinger Tebodin [6]. A soft penalty is applied on the final cost to take into account excessive deflection. This helps the algorithm to avoid iterating in the design space which give very high deflections.

Objective 2: Fabrication Efficiency Indicator

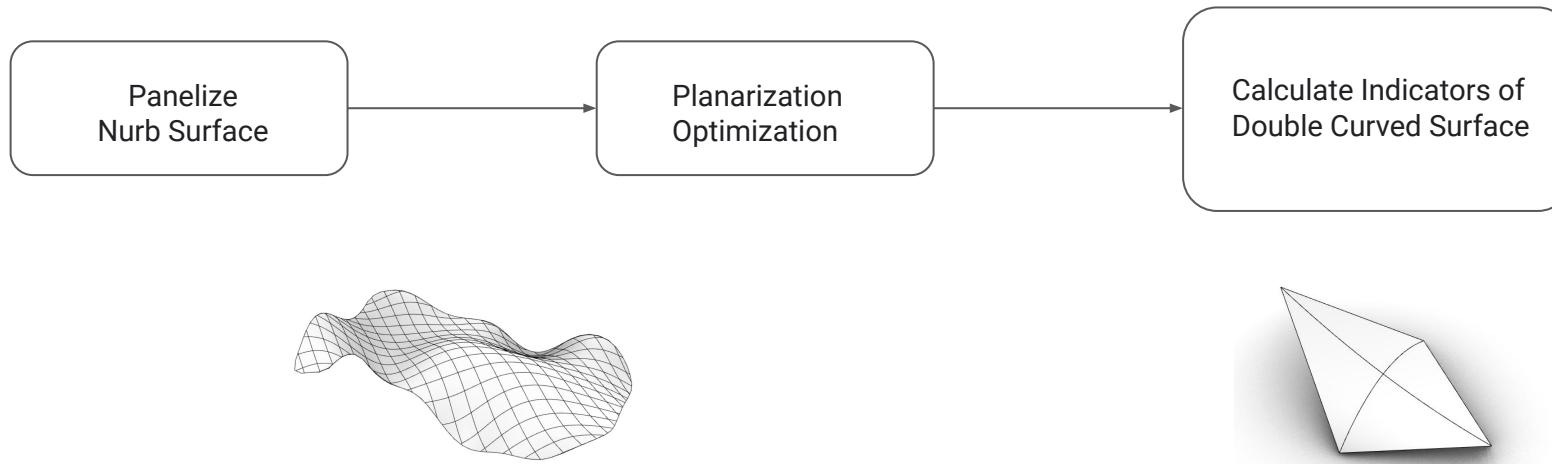


Fig. 15. Summary of the evaluation procedure for fabrication efficiency indicator.

The fabrication efficiency indicator is based on the assumption that the shell structure could be either made of precast panels or using in-situ concrete pour. Regardless of the method one prime factor which controls the cost during construction is the curvature of the shuttering panels or the precat panels. For this reason we first discretised the shell into diamond shaped panels and evaluated the flatness of each of these panels. The cumulative flatness is then reported as the value for Objective 2. Higher the value of this number indicates higher cumulative curvature and hence higher is the fabrication/construction cost. For this reason it has been termed as fabrication efficiency indicator.

II. RESULTS

- SOO
- MOO



BENCHMARK RESULTS

Objective - Material Cost

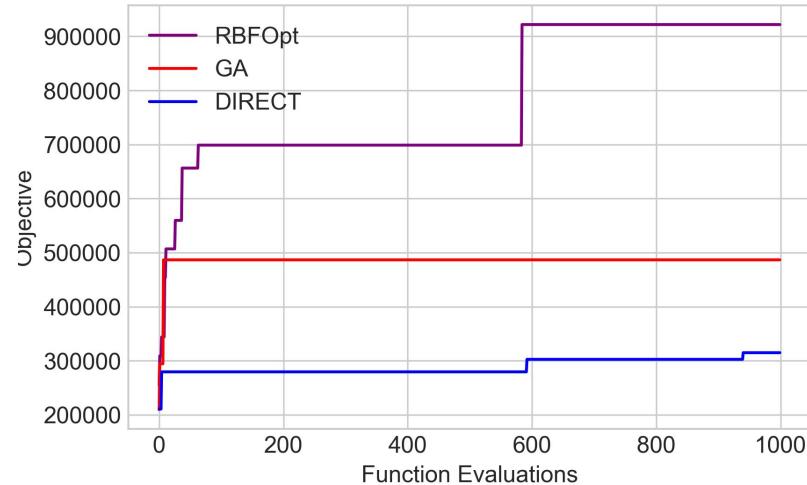


Fig. 16. Convergence graph of the three chosen single objective optimisation algorithms.

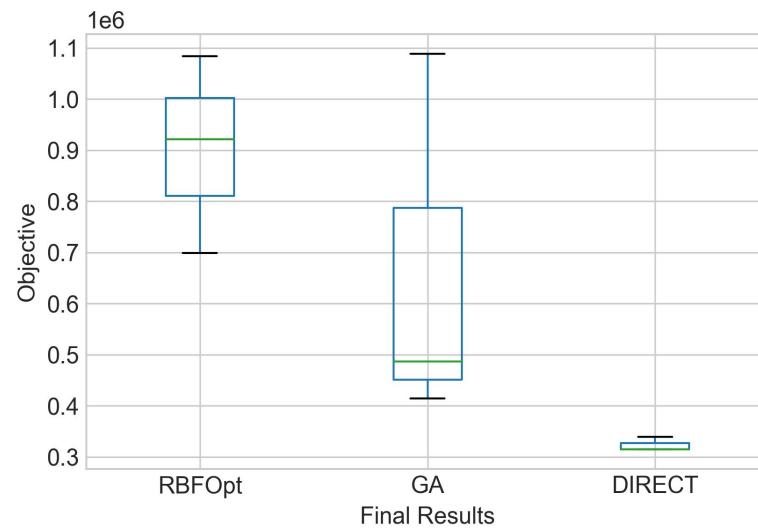
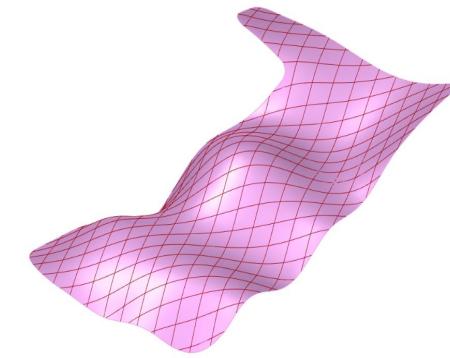
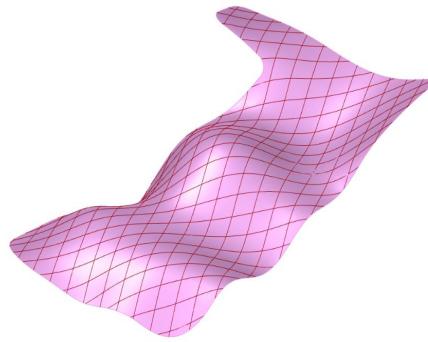
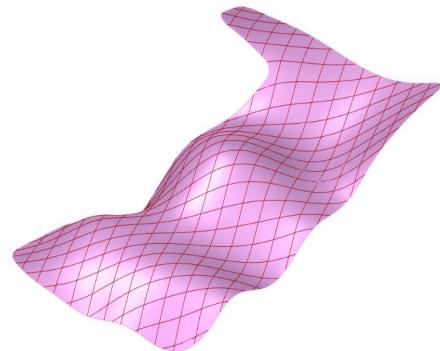


Fig. 17. Robustness graph of the three chosen single objective optimisation algorithms.

From Figure 16 on the left it's quite clear that both RBFOpt and GA are better in estimating the hypervolume even after 1000 iterations. DIRECT algorithm although converges earlier but does not converge to a result which is as optimised as others. When we look at the robustness graph, GA is not as robust as others. DIRECT is the most robust followed by RBFOpt. This result is as one would expect, since GA is a metaheuristic algorithm and DIRECT is a direct search algorithm.

BENCHMARK RESULTS

Objective - Material Cost



RBF Opt

Steel (kg)	Concrete (m ³)	Material Cost (€)	Thickness (cm)
16678	242	61589	7.7

DIRECT

Steel (kg)	Concrete (m ³)	Material Cost (€)	Thickness (cm)
12880	239	57172	8

GA

Steel (kg)	Concrete (m ³)	Material Cost (€)	Thickness (cm)
14062	230	56821	7.7

Above is the documented values of different SOO algorithms. They all seem to have values in the same range.

BENCHMARK RESULTS

Objective - Objectives - Material Cost & Fabrication Indicator

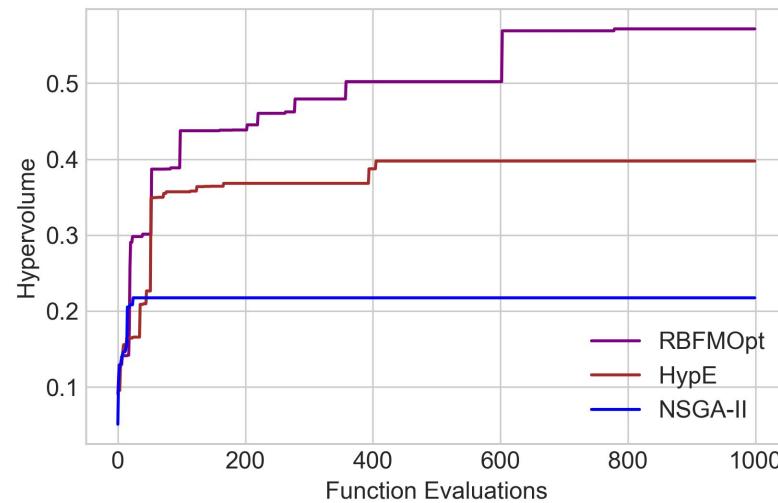


Fig. 18. Convergence graph of the three chosen single objective optimisation algorithms.

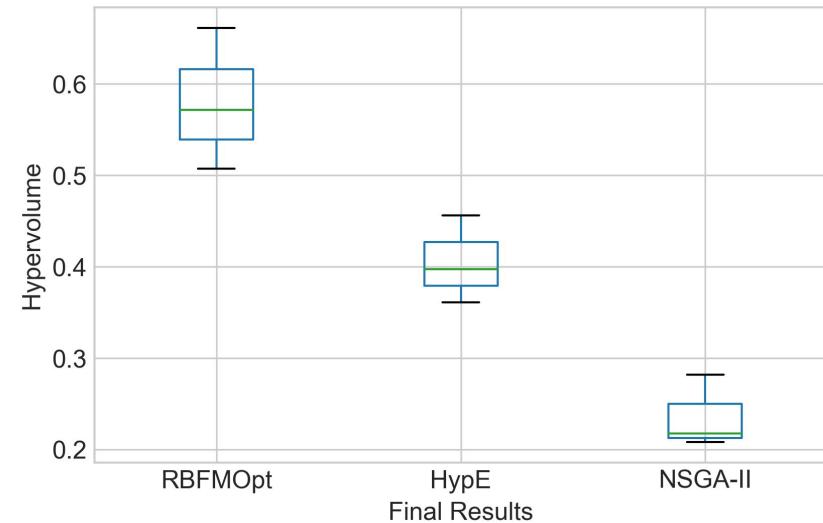


Fig. 19. Robustness graph of the three chosen single objective optimisation algorithms.

Each algorithm was run for a total 3 times each with 1000 evaluations. The convergence graph on the left shows that RBFMOpt although converges later than the other algorithms but its hypervolume approximation is the best. HypE is better than NSGA II. This is quite understandable considering NSGA II is a metaheuristic algorithm. Surprisingly, NSGA-II seems to be more robust than the other two although by not a big margin. These observations prove that one must look at both the convergence and robustness graph to come to a good conclusion. An algorithm showing high robustness may not be after converging to a satisfactory value even after large number of iterations.

BENCHMARK RESULTS

Objective - Objectives - Material Cost & Fabrication Indicator

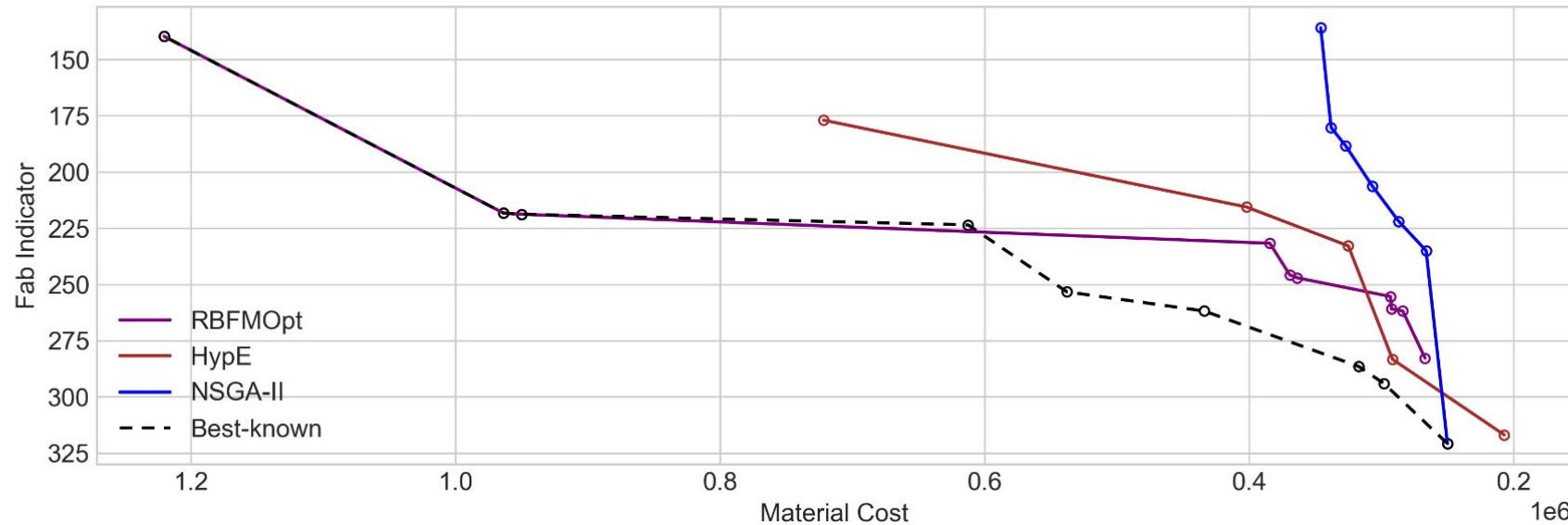


Fig. 20. Pareto front approximation for the multi-objective optimisation.

The pareto fronts plotted are pretty smooth and look complete for all the algorithms which proves that the optimization runs were successful. A clear convex pareto front indicates that the objectives are indeed conflicting strengthening the case for using Multi-Objective optimization per se. RBFMOpt as expected covers a wider range and hence heavily influences the best known front.

III. RESULTS

- Unsupervised Machine Learning



RESULTS - Pareto Rank 1, 2 & 3

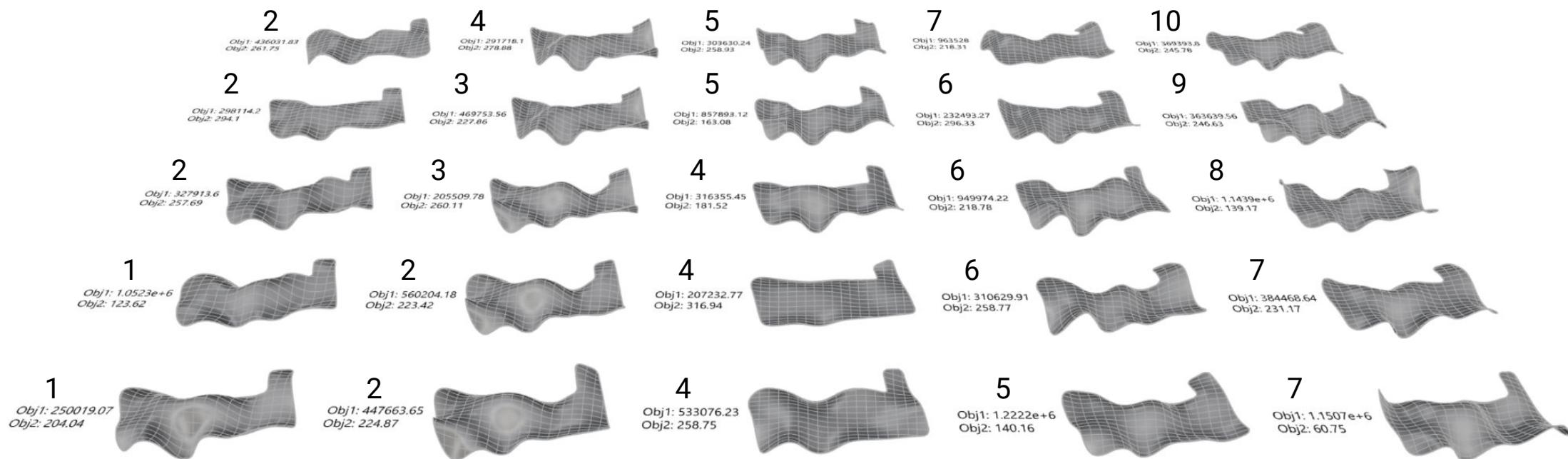


Fig. 21. All results in pareto rank 1, 2 and 3

In this all the results from pareto rank 1, 2 and 3 are plotted from all of the data generated during the MOO iterations. In the picture above they are unclustered.

CLUSTERING METHOD (without PCA)

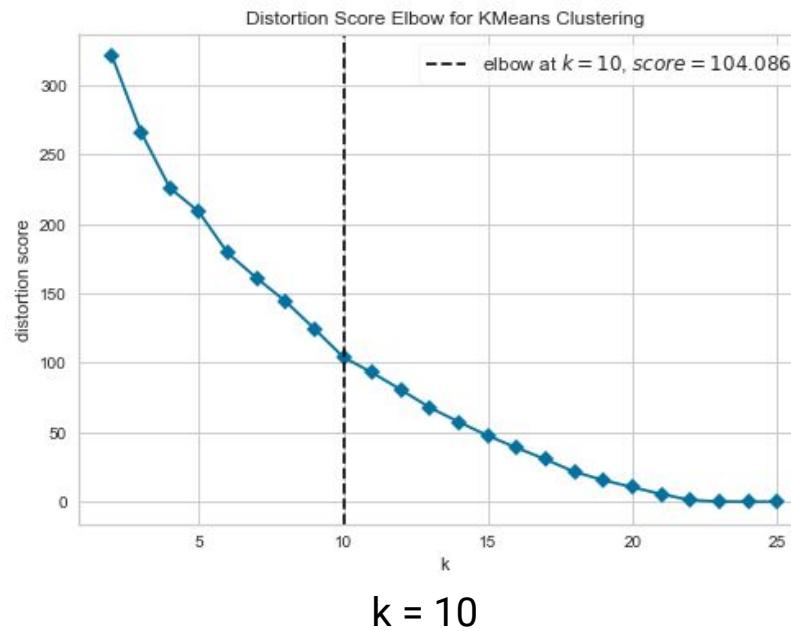


Fig. 22. The elbow method applied to the data from pareto rank 1, 2 and 3

Inertia

63.9833
51.1455
42.2504
36.0882
31.9751
28.9320
25.4834
22.8837
20.0787
17.8701
14.6058
12.3400
10.2400
9.0833
7.7331
6.1150
4.6448
3.4436
2.4976
1.6799
0.8644
0.1966
0.0158
0.0008
0.0000

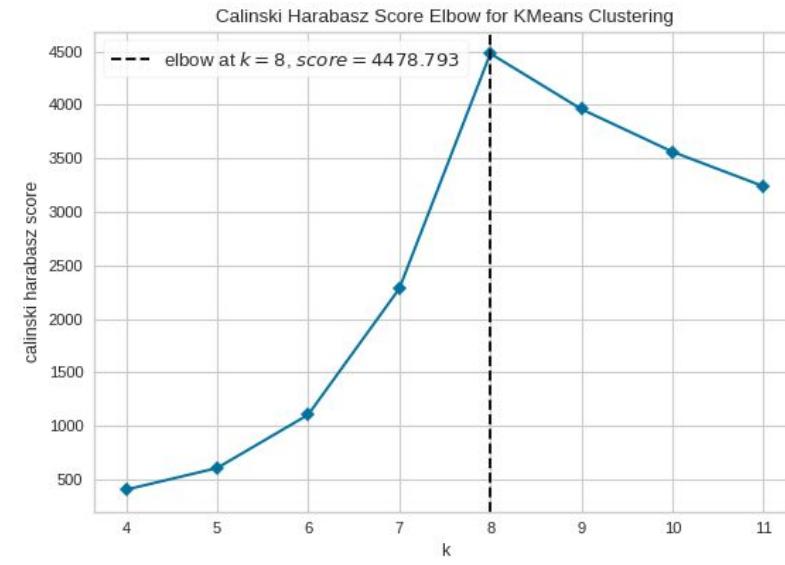


Fig. 23. The elbow method as explained in the scikit documentation. [8]

The scikit documentation states that “*It is important to remember that the “elbow” method does not work well if the data is not very clustered. In this case, you might see a smooth curve and the optimal value of K will be unclear*” [8]. Hence as can be seen in Figure 22 it is yet not very clear where the elbow occurs. We nevertheless chose to adopt the best possible result of $k = 10$.

CLUSTERING RESULTS (without PCA) - Pareto Rank 1, 2, 3

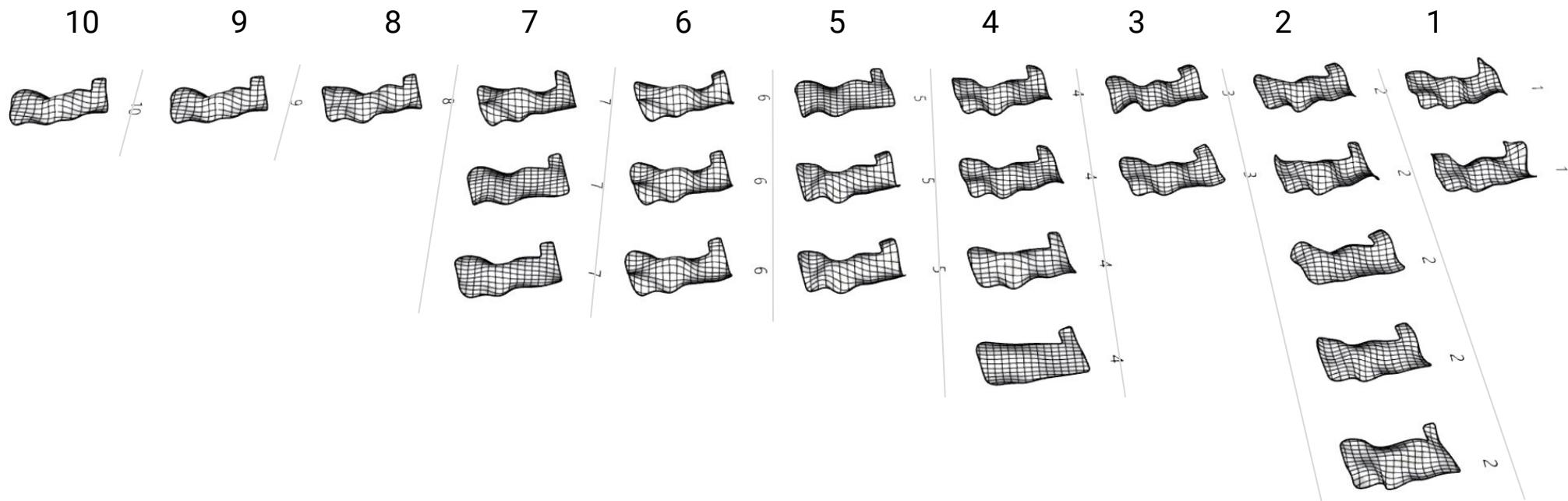


Fig. 24. Clustering results based on slab thickness and weight factors of mode shapes.

All the geometries from Pareto rank 1,2, and 3 are clustered based on parameters. Here shows the morphologically similar geometries are layout and sorted together. Looking at cluster 1 and cluster 8, 9 and 10 it seems they are mostly dominated by a single mode and the ones in between are variations of different mode shape weight factors except for one geometry in cluster 4 which seems to have been influenced by the shell thickness in that cluster.

IV. RESULTS

- Supervised Machine Learning



DATA ANALYSIS

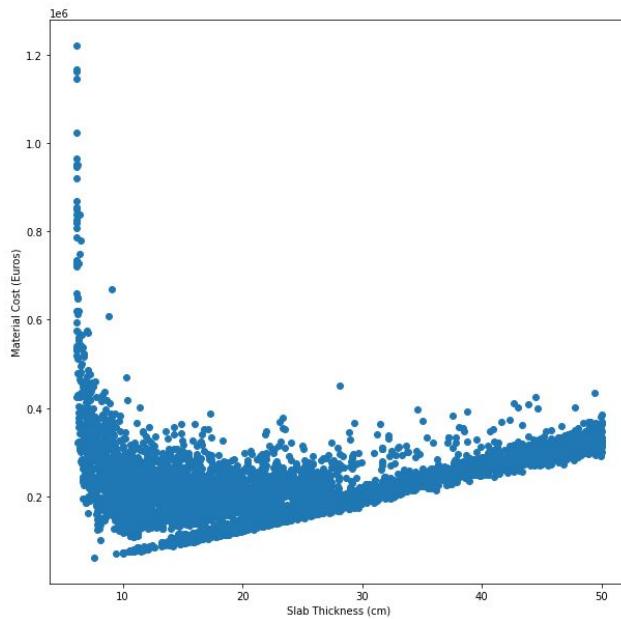


Fig. 25. Plot of Material Cost v/s Slab Thickness

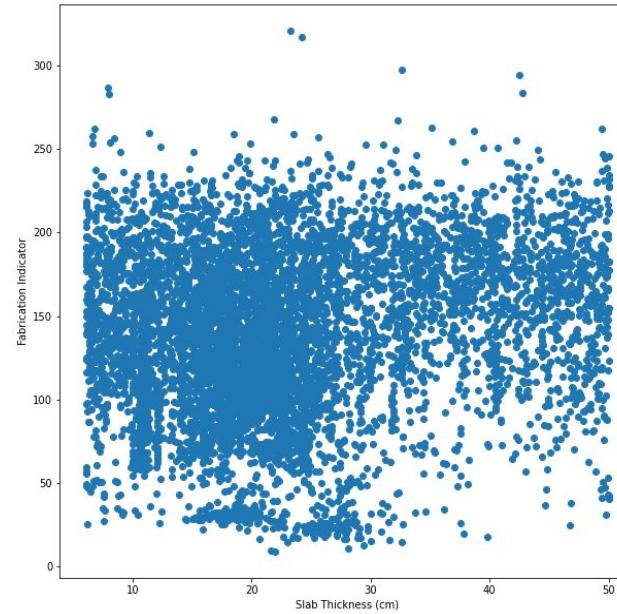


Fig. 26. Plot of Fabrication Indicator v/s Slab Thickness

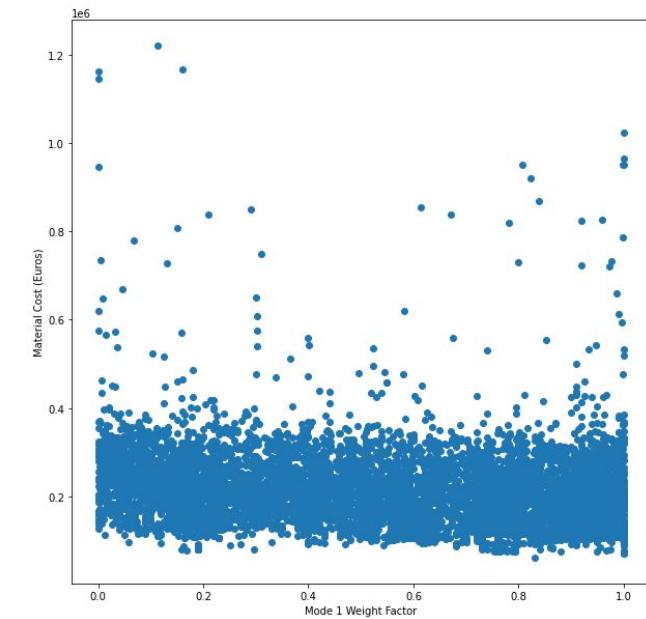


Fig. 27. Plot of Material Cost v/s Mode 1 Weight factor

The plot of material cost v/s slab thickness shows a clear convex relationship and suggests a good training could be achieved with a polynomial regression model. The plot of fabrication indicator v/s slab thickness shows a highly scattered plot and does seem to have a good corelation. We also plotted the mode shape weight factor against the material cost. agin not a very clear corelation is seen but is comparatively less scattered compared to the previous plot. This suggests that the influence of mode shape 1 is quite consistent in most of the iterations and one can use a linear model to fairly predict the values.

TRAINING DATASET

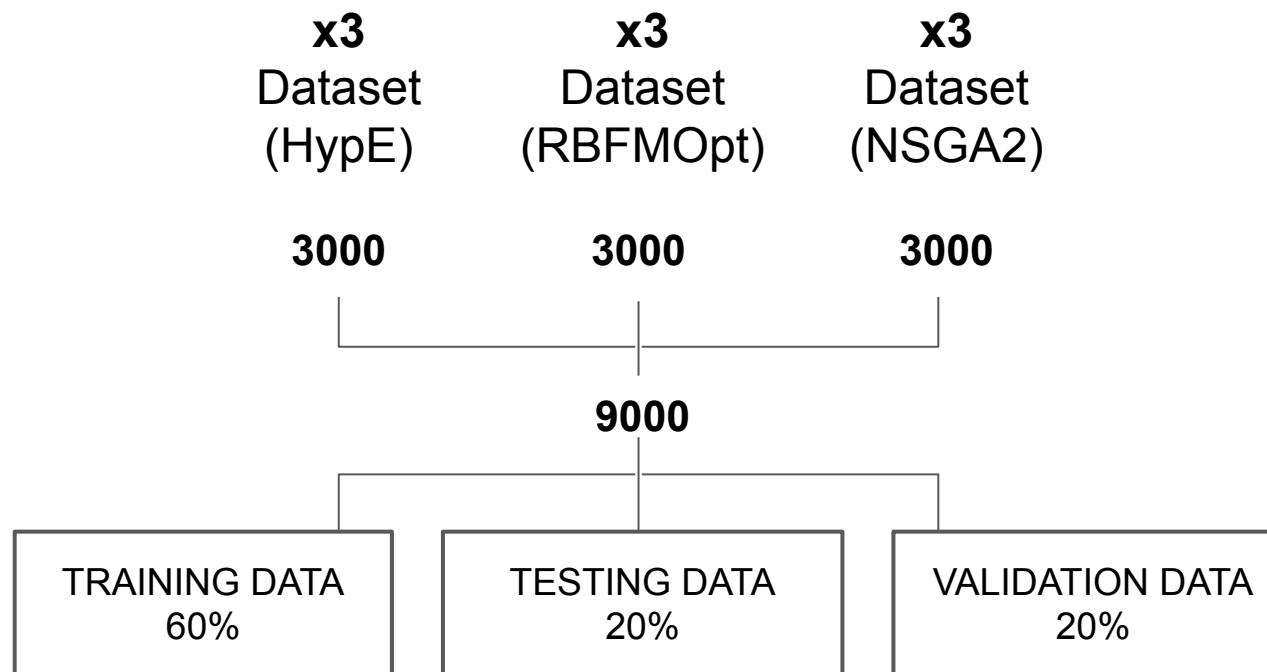


Fig. 27. Data split for model training performance evaluation.

We used all the datasets generated during the MOO iterations and split them such that we get 60% for training and 20% each for testing and validation. We used the cross validation scheme to evaluate all our models.

MODEL PERFORMANCE

Table 1: CV scores for Polynomial regression models

MODEL	CV SCORE
Polynomial Linear Regression (degree=1)	0.59
Polynomial Linear Regression (degree=2)	0.78
Polynomial Linear Regression(degree=3)	0.80

We tried Polynomial regression models with various degrees and support vector regression models using both poly and rbf kernel with various values of c and epsilon. The polynomial linear regression model of degree 3 clearly stands out. We were unable to calculate CV scores for higher degrees of polynomial regression due to serious computational limitations.

Table 2: CV scores for support vector regression models

MODEL	CV SCORE
Support Vector Regression (poly, degree 1) C=100, epsilon = 1	0.54
Support Vector Regression (poly, degree 2) C=1000, epsilon = 1	0.51
Support Vector Regression (poly, degree 3) C=1000, epsilon = 1	0.54
Support Vector Regression (rbf, degree 1) C=100, epsilon = 1	0.55
Support Vector Regression (rbf, degree 2) C=10000, epsilon = 1	0.75
Support Vector Regression (rbf, degree 3) C=10000, epsilon = 1	0.75

SUPERVISED ML

Simulated Value vs. Prediction Value

Model: Polynomial Regression (Degree 3)

Table 3: Predicted values using Polynomial regression model

SLAB THICKNESS	GROUND TRUTH	PREDICTION	PERCENTAGE DIFFERENCE
19.5 cm	Obj 1 : 175470.07 Obj 2 : 200.32	Obj 1 : 172037.42 Obj 2 : 199.76	Obj 1 : 1.96 % Obj 2 : 0.28 %
10 cm	Obj 1 : 326806.52 Obj 2 : 136.9	Obj 1 : 420399.84 Obj 2 : 134.09	Obj 1 : 28.63 % Obj 2 : 2.05 %
35.8 cm	Obj 1 : 238054.69 Obj 2 : 179.23	Obj 1 : 253518.55 Obj 2 : 179.08	Obj 1 : 6.49 % Obj 2 : 0.08 %

v. RESULTS

- Deep Learning



DEEP LEARNING

Following the previous chapter, in this chapter, the Deep Learning method has been further applied to train and build the surrogate model to predict the Fabrication Efficiency Indicator and Cost.

Models and Dataset

The datasets generated by HypE, NSGA2, and RBFMOpt Multi-Objective optimization algorithm are used as the training and testing dataset. The dataset is with 9000 data in total. The Deep Learning models will be built by using Tensorflow and Keras machine learning library.

Steps

The main processes are shown as the following:

1. Two Deep Learning models will be constructed and train with a training dataset. The DL models are built respectively with two hidden layers with 64 neurons in each layer and 2000 epochs and two hidden layers with 64 neurons in each layer and 3000 epochs.
2. Compare the MAE and LOSS and choose the model with better performance to create the HDP5 file, which has stored the trained weights.
3. Run the python script loading with the pre-trained HDP5 model and with Tensorflow running.
4. Run the grasshopper definition to get the parameters of the features.
5. Send the parameters of the features to the pre-trained model to predict.
6. The predictions value will be sent back to grasshopper environment.

DEEP LEARNING

Results Comparison: Sum of Ground Truth from Fabrication Efficiency Indicator & Cost vs. Prediction

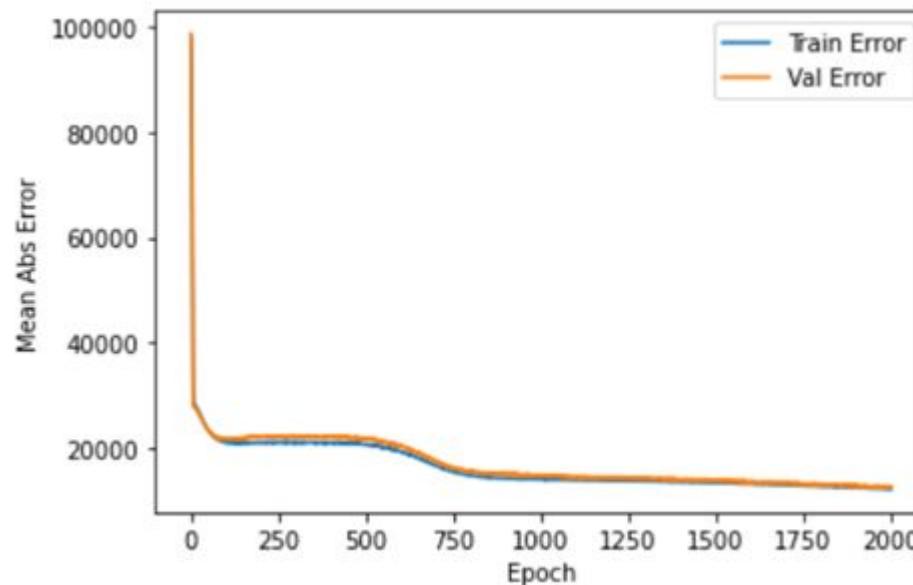


Fig. 28. Mean Abs error plot for 2000 epochs

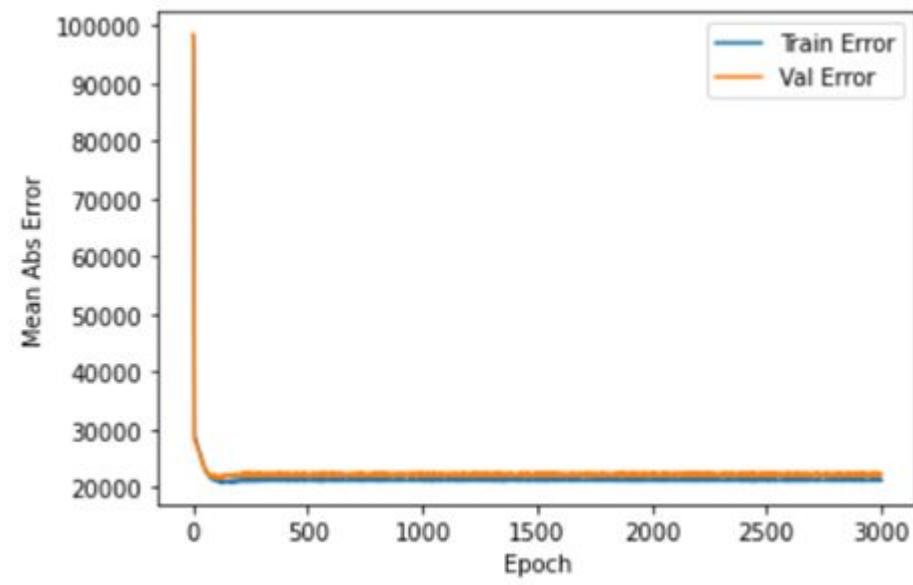


Fig. 29. Mean Abs error plot for 3000 epochs

From the graph Figure 28, it is clear that the model has start to radically reduce the error. more than 500 epochs. Both models are taking the epochs up to 2000 and 3000 epochs. So both have converged and reduced the Mean Abs Error.

DEEP LEARNING

Results Comparison: Sum of Ground Truth from Fabrication Efficiency Indicator & Cost vs. Prediction

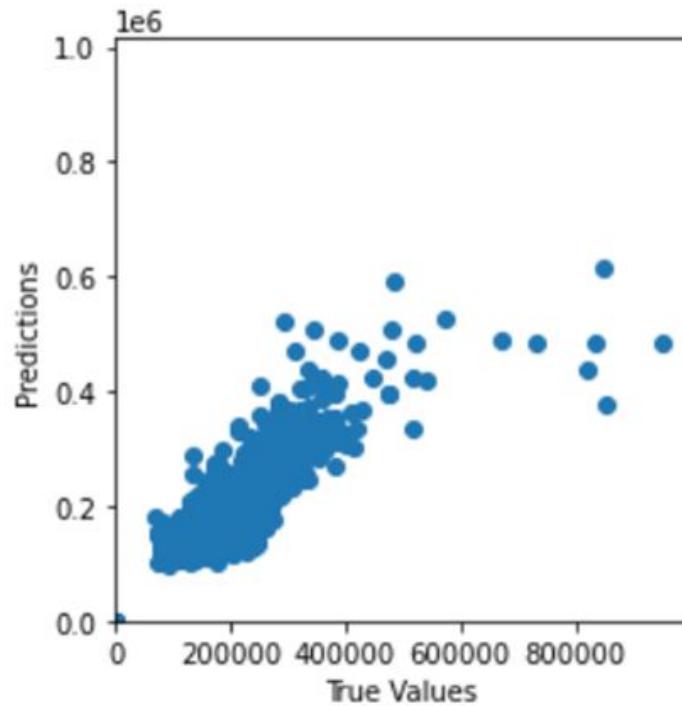


Fig. 30. Prediction v/s true values plot for 2000 epochs

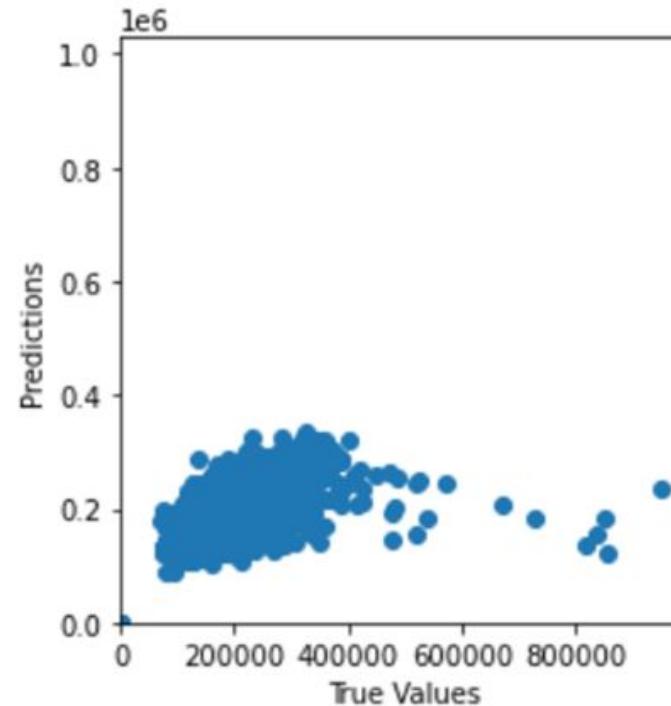


Fig. 31. Prediction v/s true values plot for 3000 epochs

From the graph Figure 30, the model with 2000 epochs shows the True Values and Predictions Value have a proportional relationship. However, the model with 3000 epochs has a less proportional relationship, i.e., increasing the number of epochs from 2000 to 3000 has decreased the prediction accuracy.

DEEP LEARNING

Predictions Accuracy

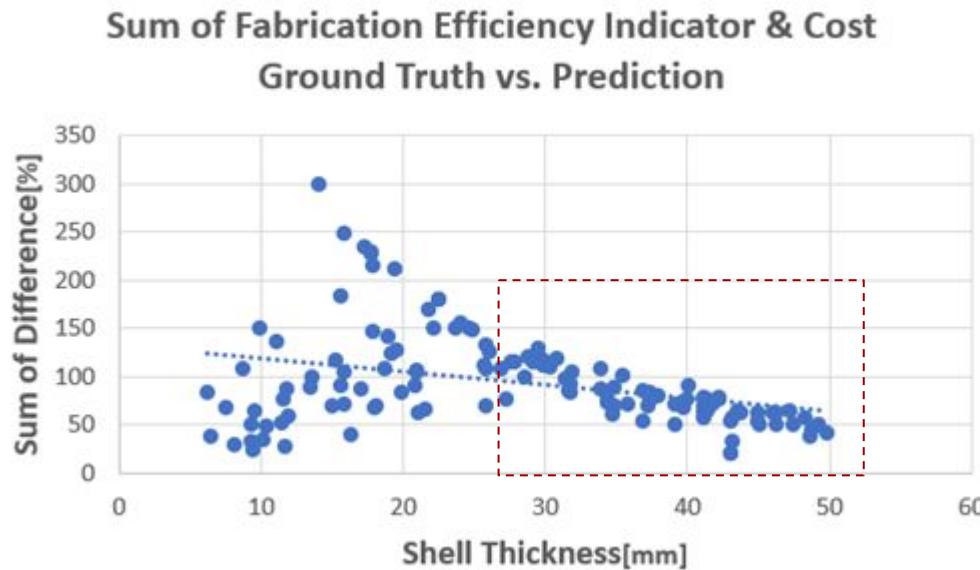
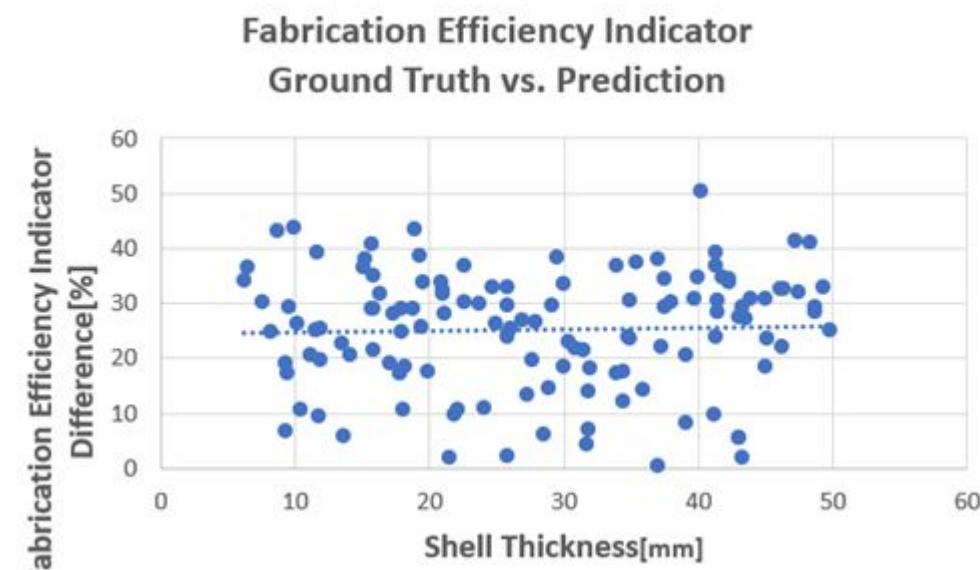


Fig. 32. Sum of Difference vs. Shell Thickness



**Fig. 33. Fabrication Efficiency Indicator Difference
vs. Shell Thickness**

The graph (Figure 32) shows the difference between the simulated and predicted values from the surrogate model trained by the deep learning method. When the shell thickness is between 30 – 50 mm, the predicted values from the surrogate model have fewer differences against the simulated value, i.e., the surrogate model in this range predicts much more accurately.

DEEP LEARNING

Comparison of Supervised Learning and Deep Learning Surrogates Models

Table 3: Predicted values using Polynomial regression model

SLAB THICKNESS	GROUND TRUTH	PREDICTION	PERCENTAGE DIFFERENCE
19.5 cm	Obj 1 : 175470 Obj 2 : 200.32	Obj 1 : 172037 Obj 2 : 199.76	Obj 1 : 1.96 % Obj 2 : 0.28 %
10 cm	Obj 1 : 326806 Obj 2 : 136.9	Obj 1 : 420399 Obj 2 : 134.09	Obj 1 : 28.63 % Obj 2 : 2.05 %
35.8 cm	Obj 1 : 238054 Obj 2 : 179.23	Obj 1 : 253518 Obj 2 : 179.08	Obj 1 : 6.49 % Obj 2 : 0.08 %

Compared to the prediction values from the Supervised Learning Surrogate and Deep Learning Surrogate models, the above tables show that Supervised Learning has a much better prediction accuracy and stable prediction results. This result is slightly different than we have expected.

The reasons why the Deep Learning surrogate model is less accurate might can be as the following reasons:

Table 4: Predicted values using DL surrogate models

SLAB THICKNESS	GROUND TRUTH	PREDICTION	PERCENTAGE DIFFERENCE
19.5 cm	Obj 1 : 327737 Obj 2 : 142.9	Obj 1 : 389832 Obj 2 : 133.8	Obj 1 : 18.94 % Obj 2 : 6.38%
10 cm	Obj 1 : 238790 Obj 2 : 165.4	Obj 1 : 368116 Obj 2 : 129.2	Obj 1 : 54.15 % Obj 2 : 21.88 %
35.8 cm	Obj 1 : 253655 Obj 2 : 142.7	Obj 1 : 388426 Obj 2 : 134.2	Obj 1 : 53.13 % Obj 2 : 5.93 %

1. Small training dataset.
2. Too less hidden layers.
3. Too little amount of neurons in each layer.
4. The small number of epochs.

To increase the training dataset, increase the hidden layers, the neurons number in each layer, and the number of epochs might help to improve the performance of the Deep Learning Model.

VI. RESULTS

1. Generative Deep Learning I: Style Transfer for Shell Substructure
2. Generative Deep Learning II: Style Transfer for Brick Facade Patterning

GENERATIVE DEEP LEARNING I : Style Transfer for Shell Substructure

Shape Optimisation using NURBS

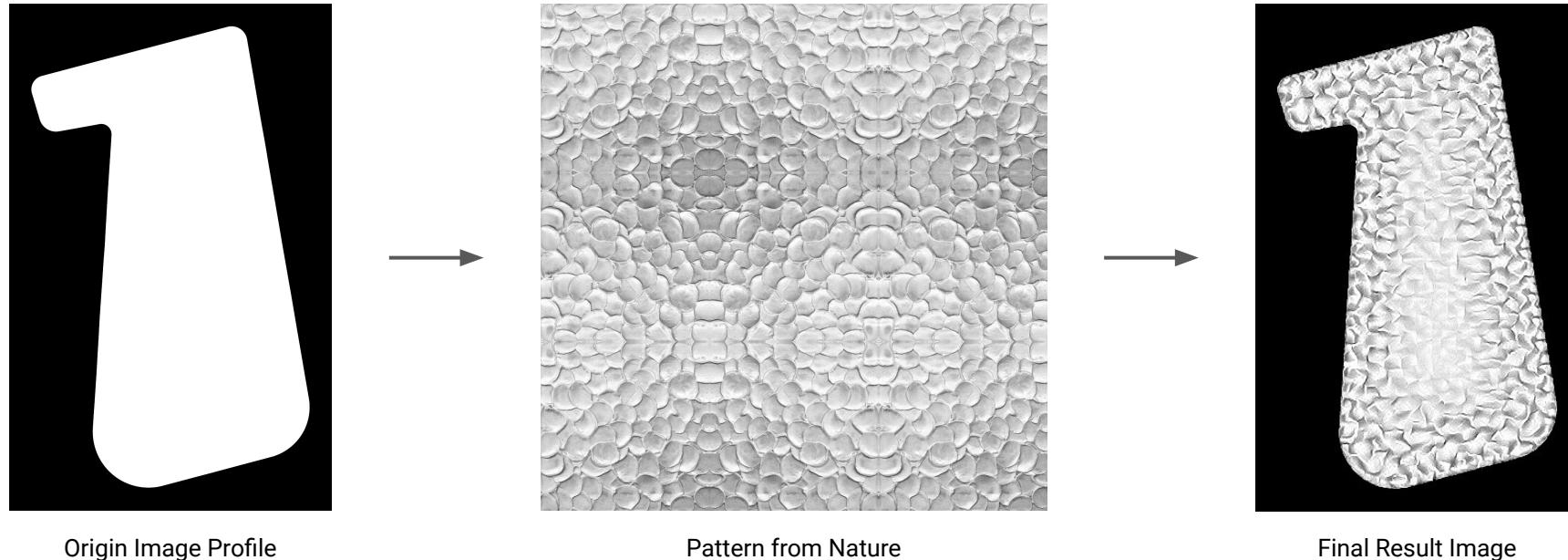


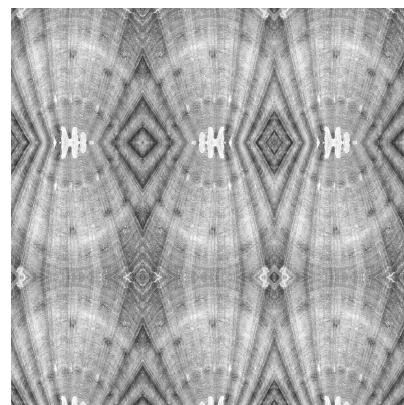
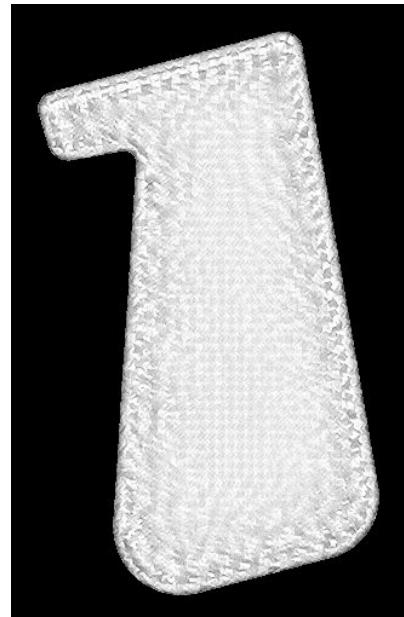
Fig. 34. Workflow of style transfer on 2D shape

The following is the workflow of the entire process

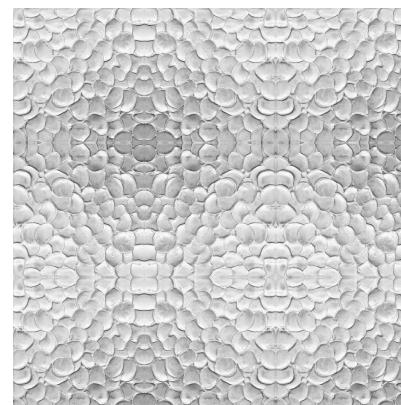
- Keep the shell with double curved panel
- Generate the pattern from algorithm
- Scattering the grid points on the surface
- Mapping the values from the style transfer image to the points
- Extrude the curve and line as ribs which stiffens the shell.
- Analyzing the structural performance
- Comparing to the original shell

GENERATIVE DEEP LEARNING I : Style Transfer for Shell Substructure

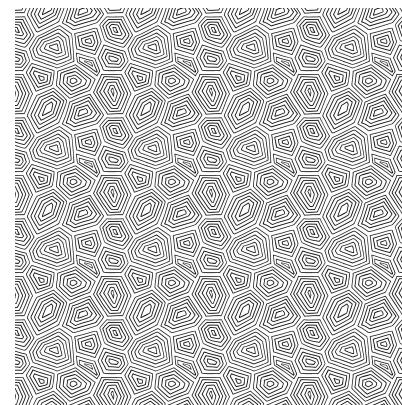
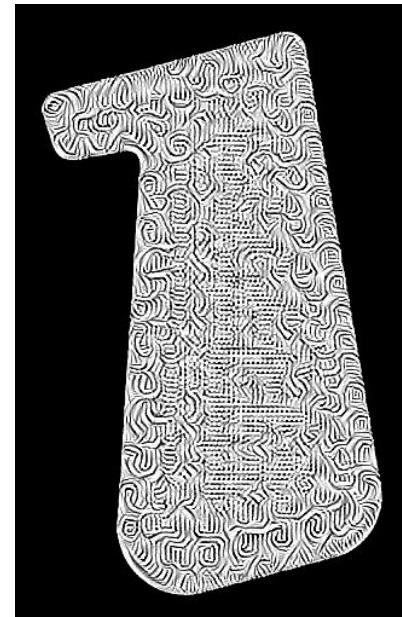
Style Transfer



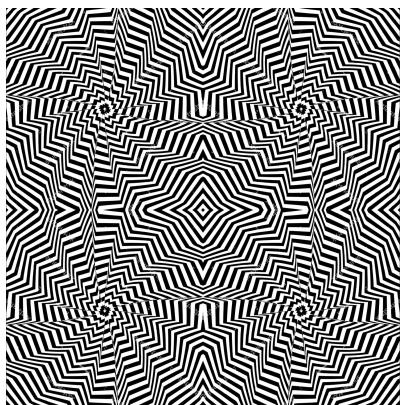
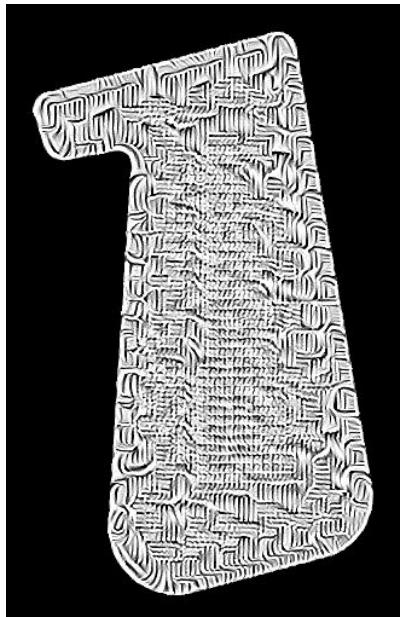
Texture 01



Texture 02



Texture 03



Texture 04

Fig. 35 Style transfer results using each of the textures.

GENERATIVE DEEP LEARNING I : Style Transfer for Shell Substructure

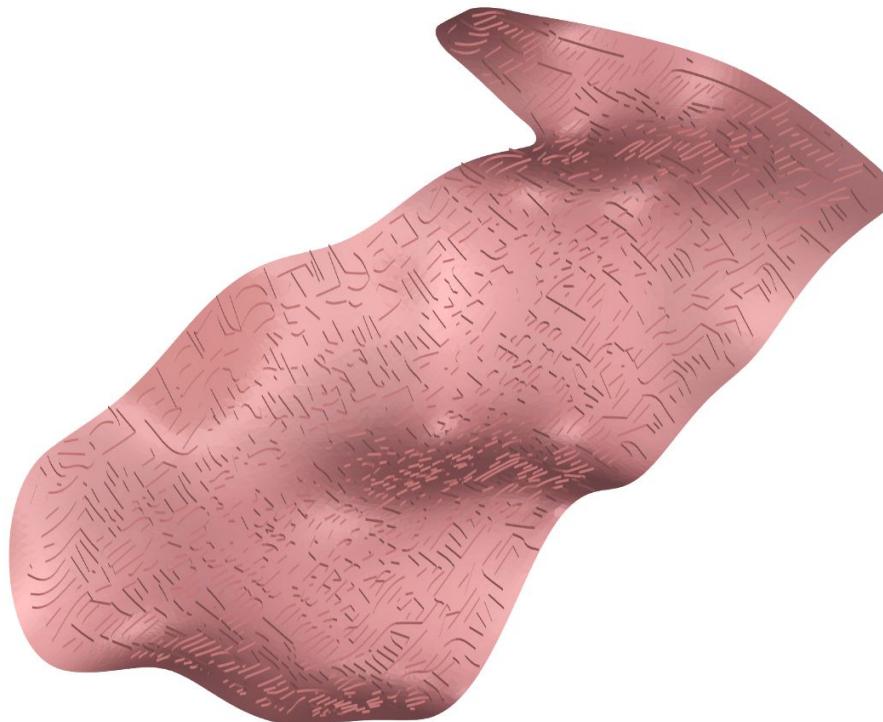


Fig. 36. 3D view Shell with ribs from Texture 04

The optimised result from SOO was taken to test this idea. Figure 36 above shows the ribs pattern on the shell and Figure 37 exactly shows how the curves and lines were placed after the style transfer in a plan view. In Table 4 we compare the results with the unstiffened shell. As one would expect the displacement is reduced albeit by a small amount. This is probably because of the additional weight introduced in to the structure because of the ribs. Nevertheless it helps and in some cases enhances the appearance of the shell.

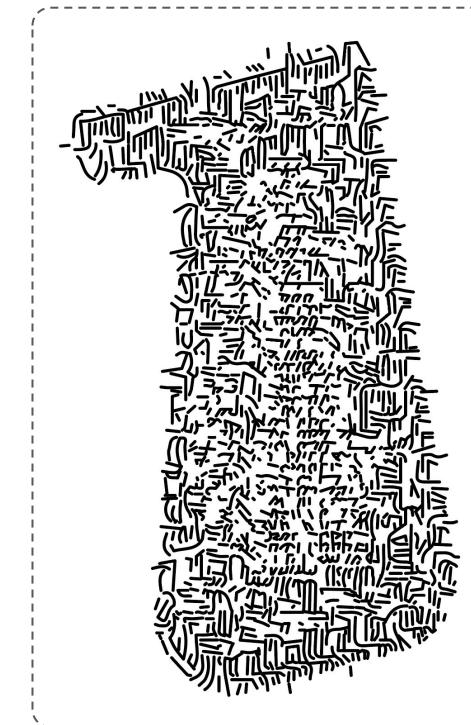


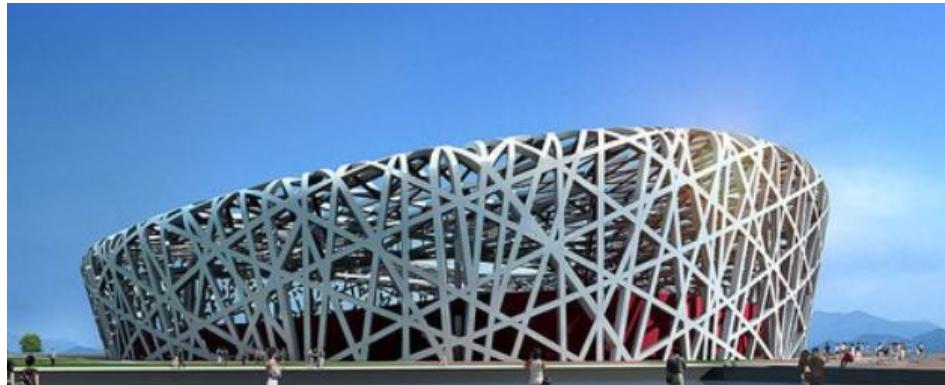
Fig. 37. Plan view Shell with ribs from Texture 04

Table 4: Comparison of structural performance.

Original Shell	Maximum Displacement (cm)	2.599757
	Elastic Energy (kN.m)	22.36022
Shell with Ribs	Maximum Displacement (cm)	2.475245
	Elastic Energy (kN.m)	29.999619

GENERATIVE DEEP LEARNING II : Brick Facade Patterning

Inspirations



Beijing National Stadium



Beijing National Water Cube

Fig. 38. Beijing National Stadium (Top) [9] and Beijing National Aquatics Center (Bottom) [10].

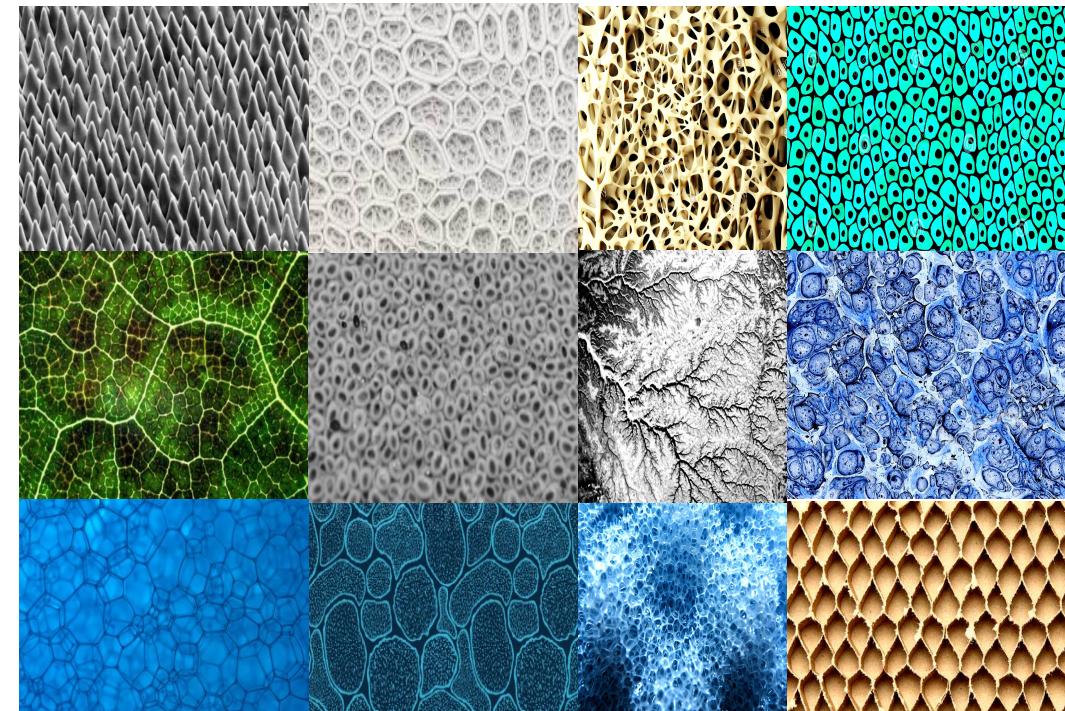


Fig. 39. Common cell texture from creatures and plantations [11].

GENERATIVE DEEP LEARNING II : Brick Facade Patterning

WORKFLOW

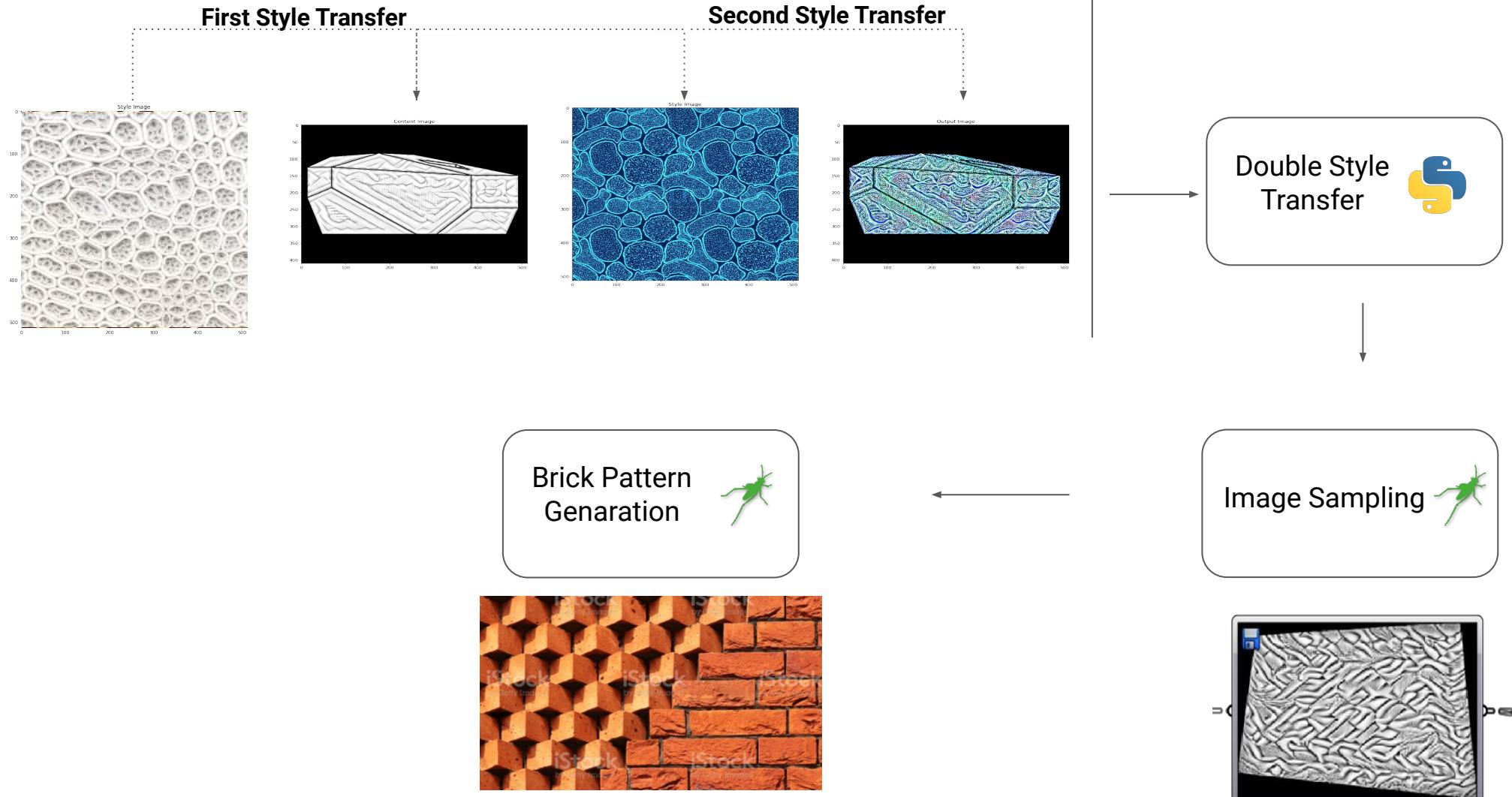


Fig. 40 Workflow of the entire process of brick facade patterning.

GENERATIVE DEEP LEARNING II : Brick Facade Patterning

WORKFLOW - STEP 2: Image Sampling

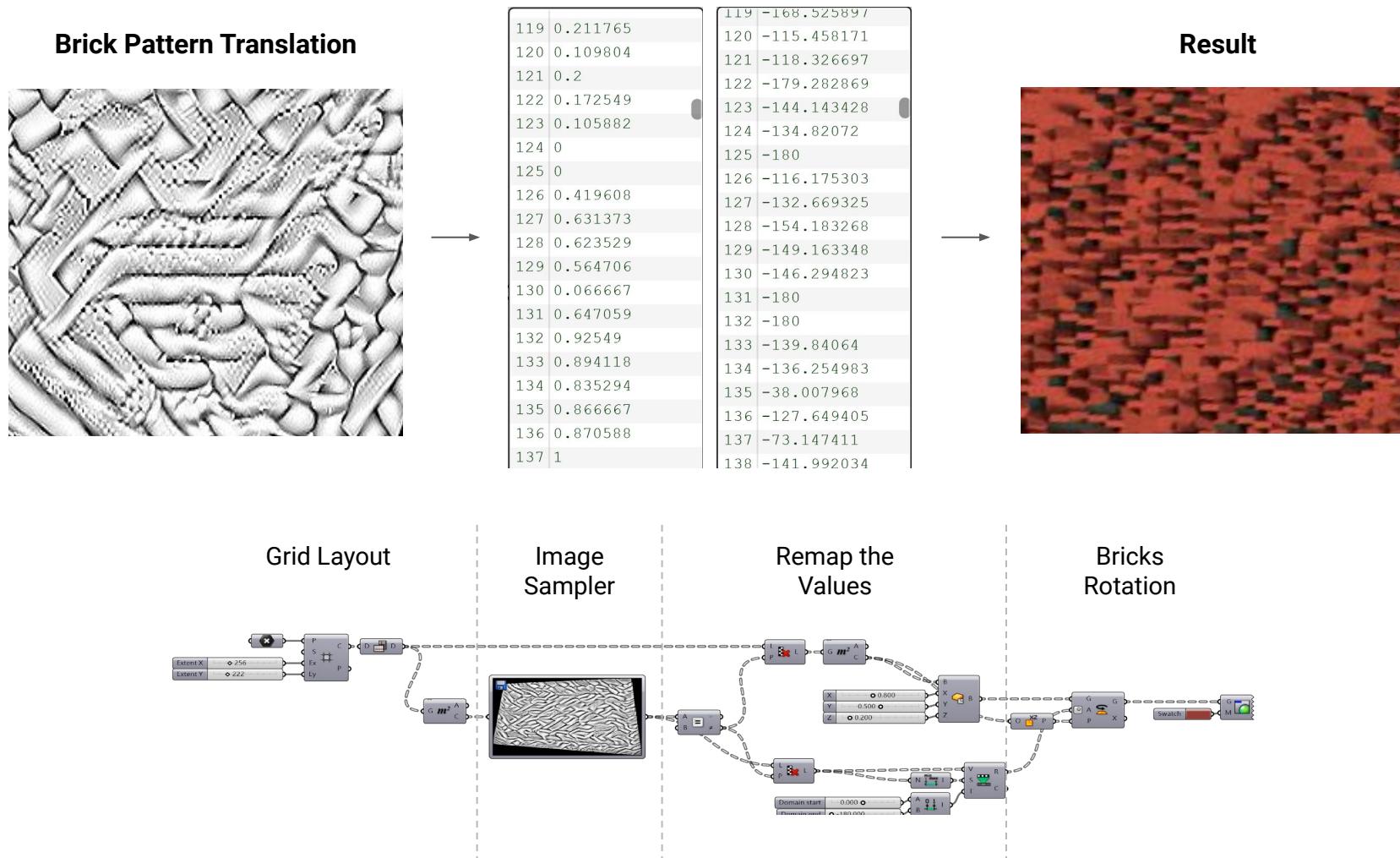


Fig. 41. Grasshopper Implementation of the brick facade patterning.

GENERATIVE DEEP LEARNING II : Brick Facade Patterning

CONTEXT APPLICATION

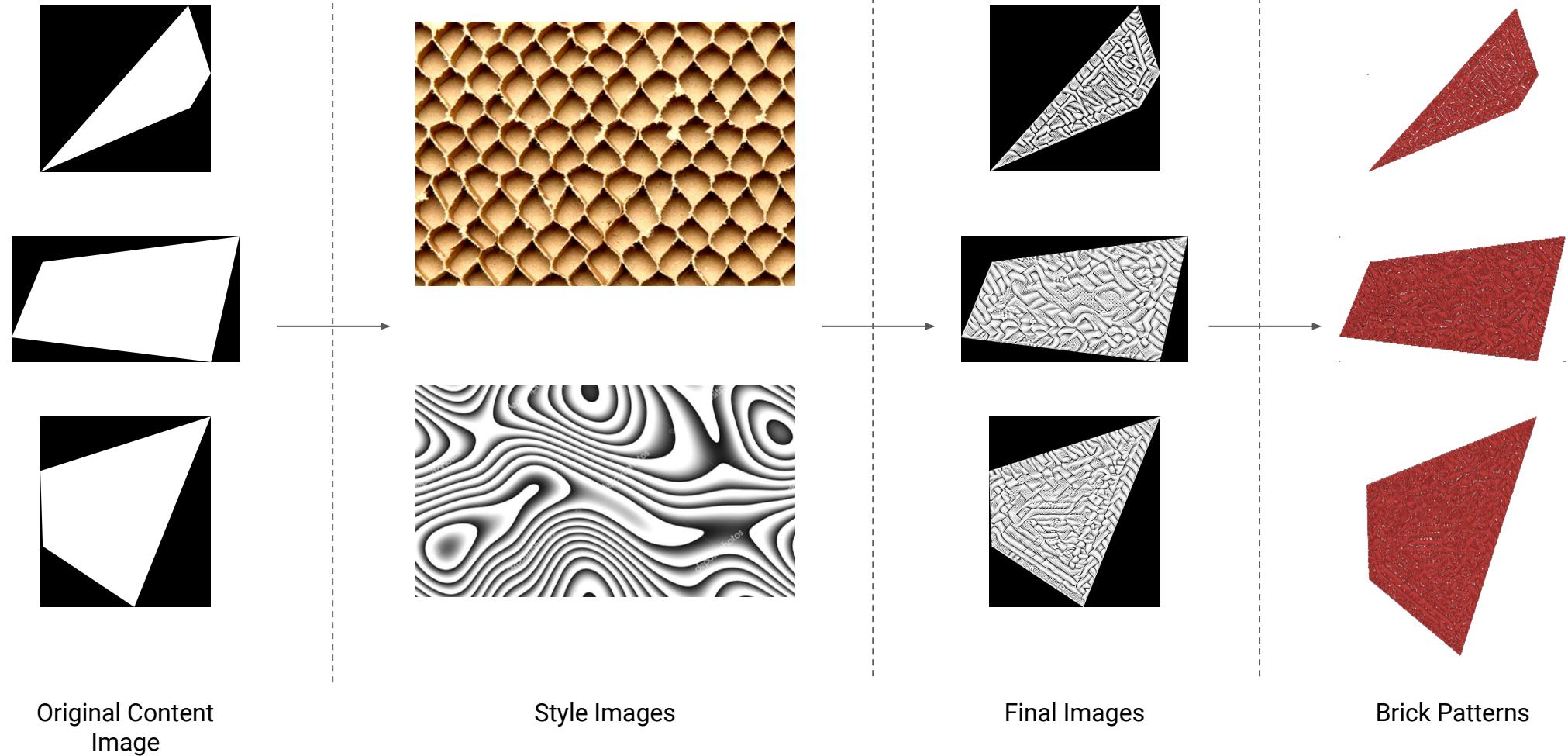


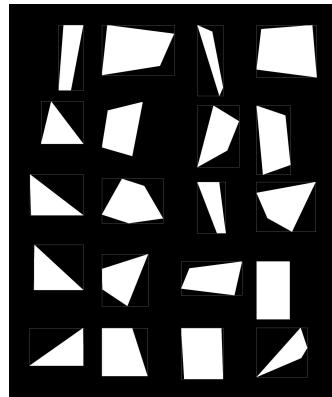
Fig. 42 Application of style transfer images on facades

GENERATIVE DEEP LEARNING II : Brick Facade Patterning

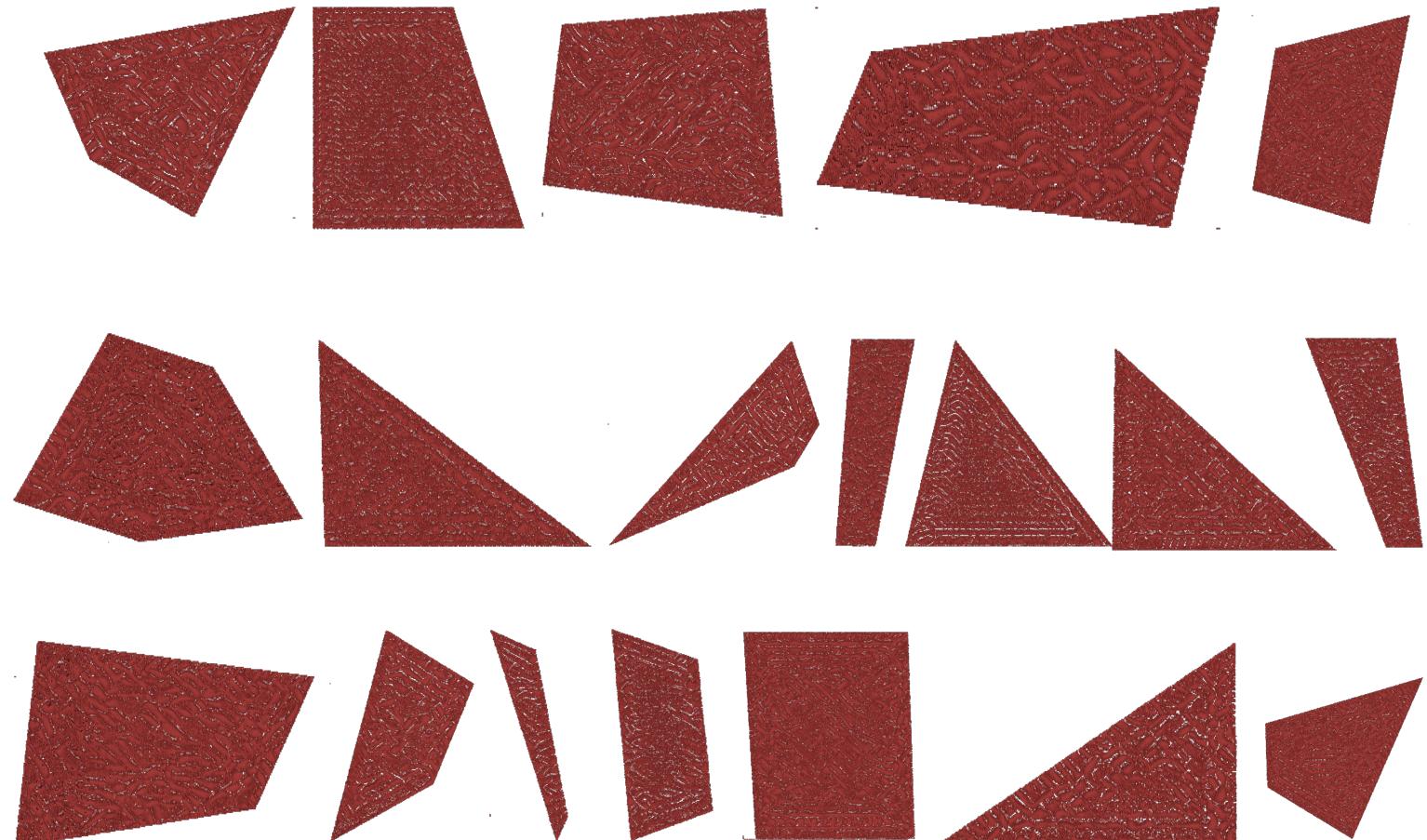
CONTEXT APPLICATION



*Casa da Musica -
Rem Koolhas*



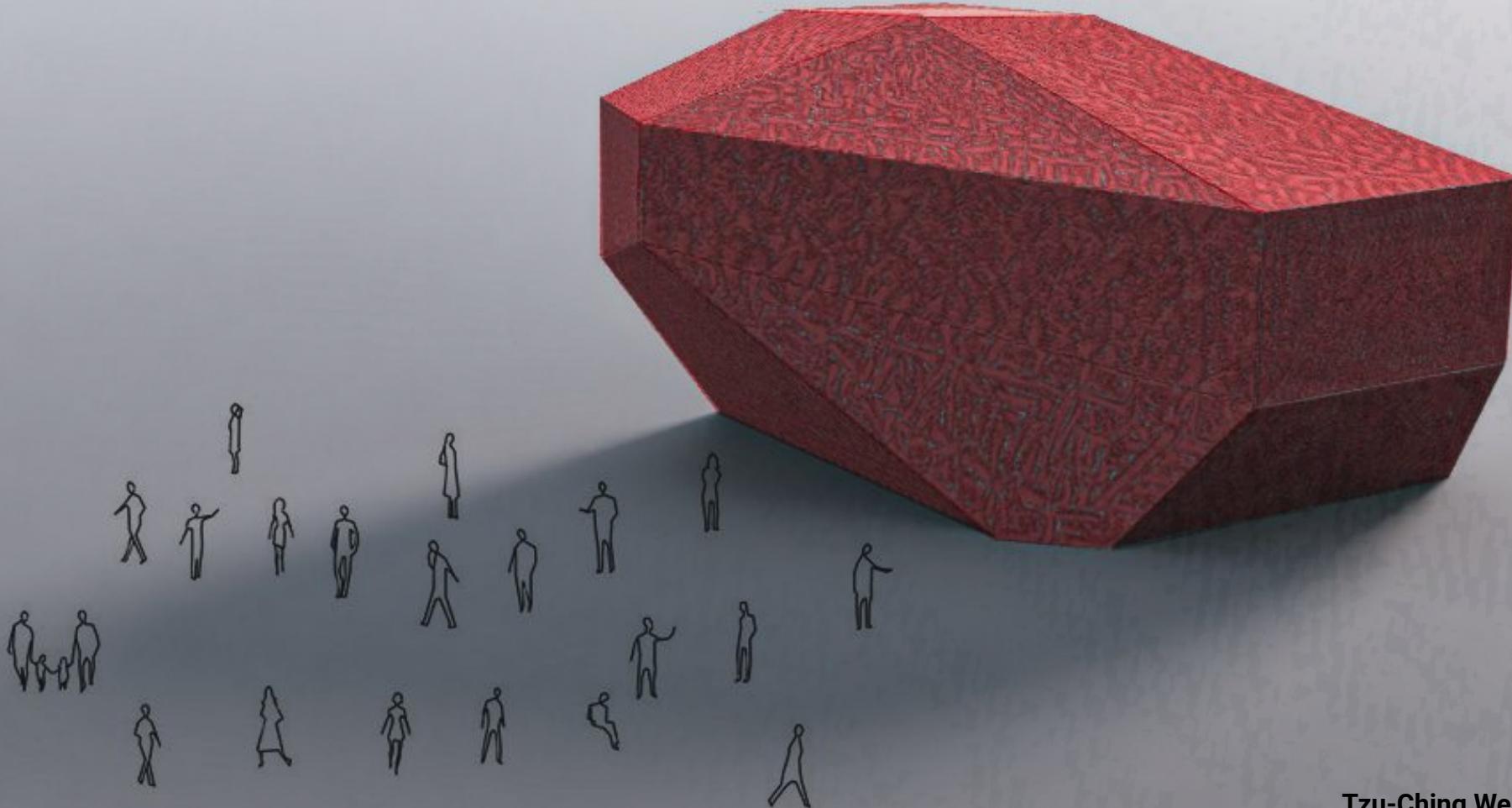
Building Faces



Building Faces with Pattern Transfer

Fig. 43. Brick pattern applied on all the facade face of Casa da Musica.

THANK YOU



Tzu-Ching Wen | Pinaki Mohanty

REFERENCES

- [1] J. Ochsendorf and P. Block, "Exploring Shell forms," in *Shell Structures for Architecture - Form Finding and Optimization*, Routledge, 2014, pp. 7-12.
- [2] D. Piker, "Kangaroo: Form Finding with Computational Physics," *Architectural Design*, pp. 136-137, 2013.
- [3] A. Pugnale and M. Sassone, "Morphogenesis and Structural Optimization of Shell Structures with the aid of a genetic algorithm.,," *Journal of the International Association for Shell and Spatial Structures*, pp. 161-166, 2007.
- [4] P. Michalatos and S. Kaijima, "Eigenshells - Structural patterns on modal forms," in *Shell Structures for Architecture: Form Finding and Optimization*, Routledge, 2014.
- [5] C. Brandt-Olsen, "Harmonic form-finding for the design of curvature-stiffened shells," University of Bath, 2015.
- [6] Bilfinger Tebodin, "Construction Cost Analysis - Industrial Projects Central and Eastern Europe," 2019.
- [7] Norihiko Dan and Associates", *ArchDaily*, 2021. [Online]. Available: <https://www.archdaily.cn/cn/771310/ri-yue-tan-you-ke-zhong-xin-norihiko-dan-and-associates>. [Accessed: 13- July- 2021].
- [8] "Elbow Method — Yellowbrick v1.3.post1 documentation", *Scikit-yb.org*, 2021. [Online]. Available: <https://www.scikit-yb.org/en/latest/api/cluster/elbow.html>. [Accessed: 13- July- 2021].
- [9]"Bird's Nest Beijing: Beijing National Stadium, How to Visit Bird's Nest, Location", *Chinabeijingprivatetour.com*, 2021. [Online]. Available: https://www.chinabeijingprivatetour.com/attractions/show/bird_s_nest.htm. [Accessed: 14- July- 2021].
- [10]"Beijing National Aquatics Center - Wikipedia", *En.wikipedia.org*, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Beijing_National_Aquatics_Center. [Accessed: 14- July- 2021].
- [11] "Common cell texture from creatures and plantations". [Online]. Available: <https://unsplash.com/s/photos/natural-texture>

TOOLS/PLUGINS

1. Modelling and Simulation Environment - Rhino/Grasshopper
2. Simulation plug-ins - Karamba
3. Optimisation Plugins - MOpussum, Goat, Octopus, Galapagos
4. Programming languages - Python in conda environment
5. Google maps