IET Software. 2018; 12(3): 251-257

Assisted surface redesign by perturbing its point cloud representation

Rafael Iván Pazos-Pérez[1], Adrian Carballal[1], Juan R. Rabuñal[2], Omar A. Mures[1], María D. García-Vidaurrázaga[3]

[1] *Department of Computation, Faculty of Computer Science, University of A Coruña, Campus de Elviña, A Coruña, Spain*
[2] *Centre of Technological Innovation in Construction and Civil Engineering (CITEEC), University of A Coruña, Campus de Elviña, A Coruña, Spain*
[3] *Department of Construction Technology, University College of Technical Architecture, University of A Coruña, Campus de Zapateira, A Coruña, Spain*

Abstract
This research study explores the use of point clouds for design geometrically complex surfaces based on genetic morphogenesis. To this end, a point-based genetic algorithm and the use of massive unstructured point clouds are proposed as a manipulation method of complex geometries. The intent of the algorithm is to improve the design experience, thus different solutions can be presented to designers. The main objective of this work is to provide examples to be adopted as user own or to help them in the creative process. This is not about providing them with a tool to 'do' the designer's creative work, but using it as a creative tool in which the user retains control of it. The powerfulness of this approach relies on the fact that the user can use any/diverse criteria (objective or subjective) to evaluate the individuals proposed as possible solutions. As part of this study, the convergence of the algorithm and the ability of diversity in the final populations of the search process will be demonstrated. Various examples of the use of the algorithm are displayed.

1 Introduction

The origins of geometrical 3D representation as it is known nowadays go back to the Renaissance period. The precursor of modern 3D visualisation was Leone Battista Alberti, who discovered the basic rules of linear perspective [1]. Alberti's findings on perspective, and Dürer's further developments [2] provided designers for the first time with representational tools aimed at radically deviating from the construction traditions and freely exploring new designs and geometries.

The next major methodological change in 3D representation took place during the age of enlightenment in post-revolutionary France, with the development of descriptive geometry. Mathematician Monge [3] radicalised geometry by developing a system for the representation of complex geometry based on orthogonal projections, combining mathematical calculations and visualisation, to allow for orthographic representation of complex curved trajectories, setting the basis for modern visualisation of complex surfaces and the orthogonal projection system [4]. Drawings as a codified method of representation were not standardised until the beginning of the nineteenth century, when Durand [5], applying Monge's findings, developed a system of orthogonal projections based on a Cartesian grid [4], further contributing to the differentiation between designers and fabricators.

It is not until the late 1990s that 3D representation and fabrication experiences a new and intense transformation due to the introduction of computers. Software packages exclusively developed for design introduced 3D representation and fabrication capabilities, with the main purpose of designing directly in 3D, allowing representation and fabrication to depart from the Euclidean orthogonal geometry [6]. Recently, designers have started to be interested in the possibility of using computational tools to self-generate designs. The idea is basically based on the use of algorithms and other computational processes to create designs with limited human interaction. Some current design approaches explore the idea of creating sets of rules through parametric algorithms that will evolve into a design by the manual manipulation of its parameters by the user, obtaining results not necessarily expected or previously thought by the designer.

More recent approaches have involved generative design using algorithms, making the most of the analytical potential of computers to deal with the inherent limitations of human beings [7]. The idea is to use generative algorithms to define formal structures created from a script. The term of Morphogenesis was already used by Turing [8] in 1952, where he explored the recurring numerical patterns of flowers, demonstrating with mathematical tools how flowers generated their complex geometry by self-organising processes, for such a purpose he idealised cells as points for the mathematical simulations. Turing's idea was simple, complex patterns can emerge from very simple rules, by the local interactions of simple unconnected elements to create complex self-organised systems [9]. Morphogenesis focuses on bottom-up logic of form finding, emphasising performance over appearance [10]. The generation of open forms using scripts should be defined as topological instead of geometrical, as it is the relation of its parts that defines the whole [11]. Morphogenesis can also be achieved through embryo genetic processes, through replication and evolutionary elimination [12]. Biological growth is a selective system, which contains generators of diversity, such as mutations or synaptic changes that are unpredictable.

Representations such as point clouds seem ideal for genetic growth compared to non-topological geometries, as they are more closely related to the principles of self-organised systems, and bottom up logic, as the cells in an organism, or the binary elements of a computer. The use of point clouds will be further explored for genetic morphogenesis of complex geometry.

2 Genetic-based morphogenesis approach

Our interest also covers further geometrical manipulation of the original object. The generation of deformations on a surface is explored, and in particular the ability to generate diversity, by creating families resulting from the manipulation of the original shape. To this end, in this paper an evolutionary morphogenesis process is proposed to introduce the alterations on the basis surface and to generate a diversity of similar solutions. The idea is that from the point-based geometry of an existing surface as a starting point, some alterations on its surface not manually but by a morphogenesis process are generated. In addition, we think preferable to automatically generate great diversity of similar samples, without equal solutions from the original 3D geometry.

Previous works [13, 14] studied about computing with geometric constraints such as the use of force of equilibrium, cost or other structural constraints. In particular, software such as EvoluteTool or Galapagos is oriented to this type of problematic, in which designed structures are built from simple elements. Other works in the field deals with the combination of form and fabrication as Fraternali [15] in masonry.

In this paper, a genetic algorithm is proposed [16, 17], which alters the appearance of the surface modifying a previously given point-cloud representation to obtain the new design candidates. Evolution is probably one of the best search processes till date and has been performed in nature for millennia with a clear ‘worth’, so its application to the creative design field seems more than reasonable and is widespread [17–19]. The novelty of the algorithm proposed is that it uses mathematical expressions in order to create a natural texture from a point-cloud representation of the original surface. Unlike Rhinoceros, Grasshopper or other parametric software packages, it is not the user who decides how the surface gets deformed, but the expressions. In fact, the user simply decides which of the examples offered by the system are consistent with his/her criteria, just as design evaluators [20, 21].

This interaction between human and computer has some advantages, especially in the creative process. The user chooses following the criteria they deem appropriate without ‘wasting’ time in the execution process, which tends to be more costly in terms of time. The user evaluates the individuals guiding the evolution process. The powerfulness of this approach relies in the fact that the user can use any/diverse criteria (objective or subjective) to evaluate the individuals proposed as possible solutions.

2.1 Surface redesign algorithm based on genetic morphogenesis deformations

The developed genetic algorithm imitates the mechanisms of natural selection: survival of the fittest (selection), recombination of the most promising genetic material (crossover) with slight modifications (mutation). The intent of the algorithm is to improve the design experience, thus different solutions are presented to designers to be adopted as their own or to help them in the creative process.

2.2 Sampling a surface

Given an initial free-form surface or polysurface a uniform sampling in the u, v domain in needed. First, it must be determined the number of surfaces of the initial surface, since the framework supports both surfaces and polysurfaces. The next step is the calculation of the number of points that are needed to cover each surface. For this purpose, a calculation of how many points per unit (PPU) are used is carried out. Once this quantity is established the number of U_{steps} and V_{steps} is easily computed with the next equations:

$$U_{steps} = (U_{max} - U_{min}) * \text{PPU} \tag{1}$$

$$V_{steps} = (V_{max} - V_{min}) * \text{PPU} \tag{2}$$

For polysurfaces, the U_{steps} and V_{steps} values from the less dense surface should be used. After computing the number of steps necessary for each surface, the next stage consists on point sampling based on uniformly distributed lines [22] (see Fig. 1 as an example sphere).

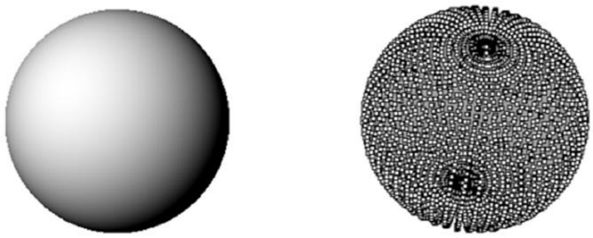

Fig. 1 *Point cloud sampling from sphere*

3 Evolving point-cloud representation surfaces

The idea is to alter the appearance of an initial surface modifying a previously given point cloud representation to obtain the new design candidates. The algorithm is based on interactive evolutionary computation (IEC) [20], imitating the evolution process in terms of selection/reproduction. Selection will guarantee that the most adapted will survive for reproduction, whilst ensures the inheritance of the fittest genetic material over descents. It can also be considered a search process to identify better individuals in the space of all possible individuals. The user evaluates the individuals guiding the evolution process.

3.1 Gene representation

The genotype of the individuals is a mathematical expression represented by a tree. The trees are constructed from a lexicon of binary functions as tree nodes and terminals as leaves. The terminal set is composed of the variables *x* and *y* of a Cartesian model. This idea of morphogenesis deformation is based on NEvAr developed by Machado and Cardoso [23, 24], but applied to 3D surface design instead of image design. The interpretation of a genotype (an individual) results in a phenotype, which in NEvAr's case was an image. To generate an image, they evaluate the expression for each pixel coordinate and the output is interpreted as the greyscale value of the corresponding pixel [23].

By contrast, the algorithm uses point cloud with a predetermined density that represents the initial geometry as input data. Once obtained, some modifications are performed directly on such point cloud, so that the value obtained by evaluating the arithmetic expression on each point determines its displacement over its normal (offsets). Once completed, the new resulting geometry is reassembled (Fig. 2).

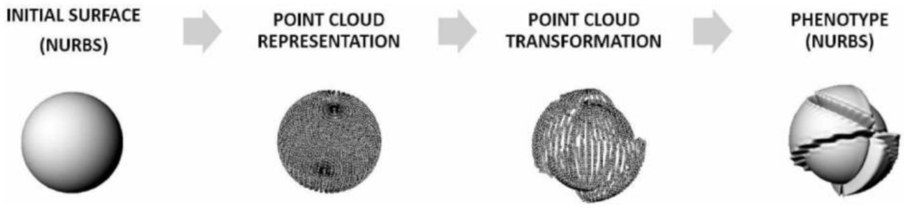

Fig. 2 *Visual genetic morphogenesis process*

A series of initially preset parameters determine how the deformations are applied, such as range of min/max of normal displacements, minimum percentage of affected surface, maximum and minimum genotype tree size of the first population above those related to the evolution process such as mutation and crossover rates. All parameters used in Fig. 2 are detailed in Table 1.

Table 1 Predefined parameters used by the genetic programming (GP) approach for the design process

Parameter	Value
point-cloud UV density	50 × 50 pps
range of displacements	[−0.1; 0.1] cm
minimum affected surface	45%
maximum tree size	40
minimum tree size	5
functions (tree nodes)	+, −, *, /
terminals (tree leaves)	‘*x*’, ‘*y*'
initialisation	ramped half and half
fitness function	binary EIC
recombination strategy	one-point crossover
mutation strategy	leave flipping
mutation rate (p_m)	[0.01; 0.15]
selection strategy	proportional Roulette wheel
replacement strategy	invert fitness

3.2 Algorithm specifications

Algorithm 1 (see Fig. 3) shows the pseudocode used. For interpretation, it is important to take into account the following input variables: population size N, population P, auxiliary population Q and the initial surface or polysurface Z_0. It is represented with Z_f a design presented by the algorithm that satisfies the user.

Functions involved in the algorithm are described below:

- *crowd_population(P, N)*: breeds new individuals in P until max population value N is reached. If P is empty, all elements are randomly created. In other case, a reproduction/mutation process creates new elements.
- *random_selection(P)*: select one individual from P using proportional roulette wheel selection method.
- *evolve_process(Z_0, parents)*: computes reproduction and mutation using both parents and the initial surface Z_0.
- *isValid(offspring)*: checks if offspring fulfil all design preconditions/constraints.
- *insert(offspring, Q)*: this function first converts the offspring to its phenotype and after is added to the auxiliary population Q.
- *user_evaluation()*: once $|Q| = N$, the user will determine delectable designs keeping those which will participate in next step.

```
Require: P ← ∅
  while Z_f ∋ P do
    crowd_population(P,N);
    Q ← ∅;
    while |Q| < N do
      parents ← random_selection(P);
      offspring ← evolve_process(Z_0, parents);
      if isValid(offspring) then
        insert(offspring,Q);
      end if
    end while
    P ← user_evaluation(Q);
  end while
  return Z_f
```

Fig. 3 *Algorithm 1: GP pseudocode*

Looking for the difference of the proposed algorithm, it has to be noticed the possibility of ‘hermaphroditism’ allowed in the roulette wheel selection method. Generally speaking, both parents are allowed to be the same individual in selection and therefore the resulting new genotype comes from the same individual. In our context, this modification is required. For example, when the user holds a lonely survival design, indicating that the design is pleasing and wants it to be exploited.

Besides, percentages of mutation (p_m) and crossover (p_c) vary depending on the percentage of designs that survive each iteration and the range of allowable values for p_m

$$p_m = \frac{N - |P|}{N} + \max(p_m) + \frac{|P|}{N} \times \min(p_m) \tag{3}$$

$$p_c = 1 - p_m \tag{4}$$

For those cases where the percentage is a very high, mutation value decreases while the crossover increases. Otherwise, if the percentage of surviving is small, then the mutation rate rises while the crossover down, trying new designs are sufficiently different from their parents. In both cases, this variability is progressive and is directly proportionate to the maximum number of people allowed in the population.

Unlike DeJong's method [25], only valid offspring whose visual difference exceeds a percentage with respect to their parents will be accepted to guarantee genotypic diversity. To simplify this method, the value of p_m is used. If the percentage of surviving individuals in a given iteration is high, it is necessary to increase the capacity of exploration. If not, what needs to be improved is the ability of exploitation. All parameters used by the GP approach were detailed in Table 1.

3.3 Convergence and diversity

Within the search process for possible solutions is important to achieve both convergence and diversity. Since the algorithm was designed for its use as IEC and the achievement of the corresponding tests of convergence would be complicated temporarily speaking, the convergence of automated form has been studied. Due to this, three graphical representations corresponding to Matyas, SixHump Camel and Easom functions [26] in the range [−1, 1] are used as goals (Fig. 4). These functions are well known and widely used in optimisation.

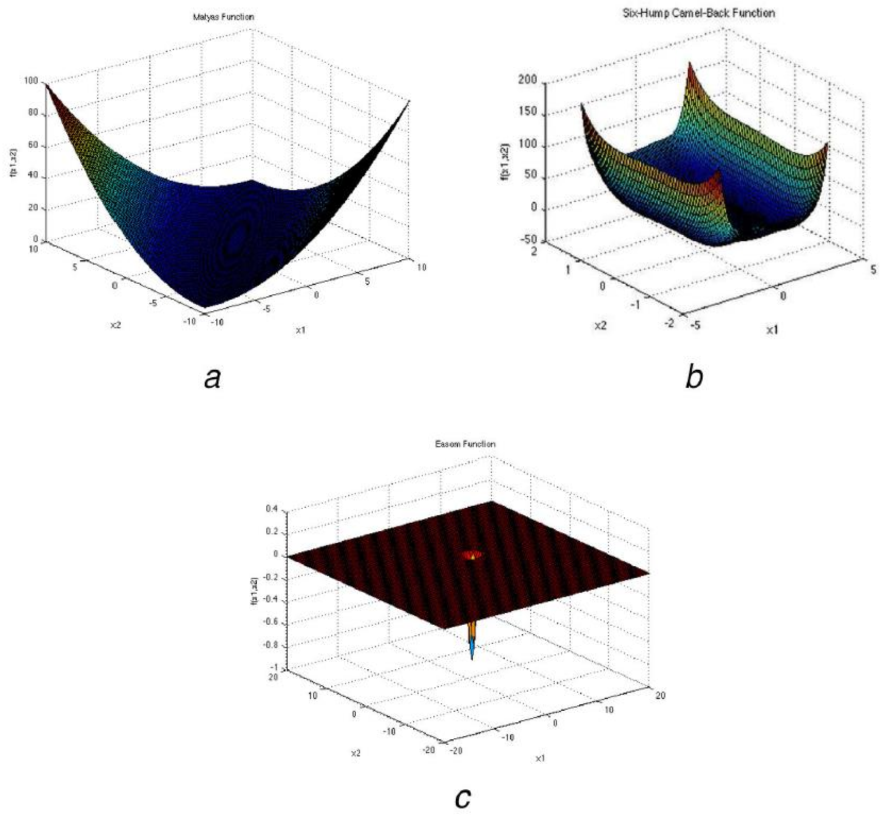

Fig. 4 *Optimisation functions used for convergence testing*
(a) Matyas (plate shaped), ***(b)*** SixHump (valley shaped), ***(c)*** Easom (steep ridges)

Instead of using a human subject to perform the selection of the individuals to survive in the evolutionary process, we opted to determine the fitness of each individual S_x with regard to the representation in point cloud of the target surface S_0 using both mean and standard deviation of every point in the cloud simultaneously.

The fitness function exponentially penalises surfaces where at least a point is significantly far from the target. In view of this, preference is given to those surfaces where the average difference between its points of representation is minimised.

To confirm the stochastic stability of the convergence, 50 independent runs for each objective surface has been carried out. Fig. 5 shows the average fitness of the best individual. Results confirm the convergence achieved with a final fitness <0.005.

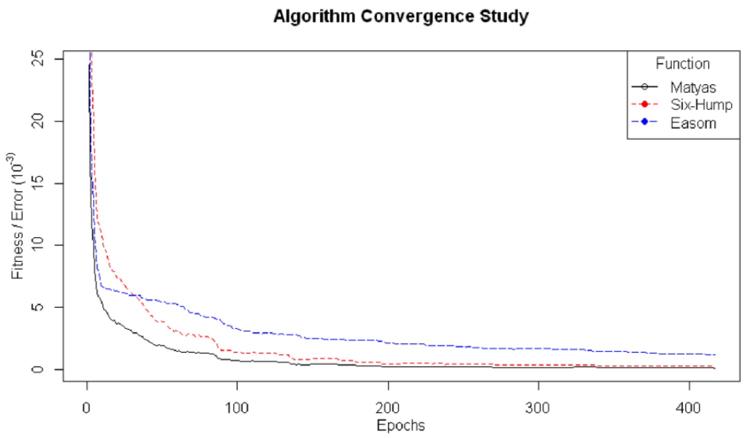

Fig. 5 *Convergence graph for the three objective functions*

In terms of diversity, it is important that visual alternatives are presented to the user. For this reason, it is essential to ensure the diversity of each population in the sense that there is no individual visually equal to another. Fig. 6 shows the differences between every individual from the final population obtained for each target function. As notice no zero-case value is reached, which would indicate that two individuals are visually identical.

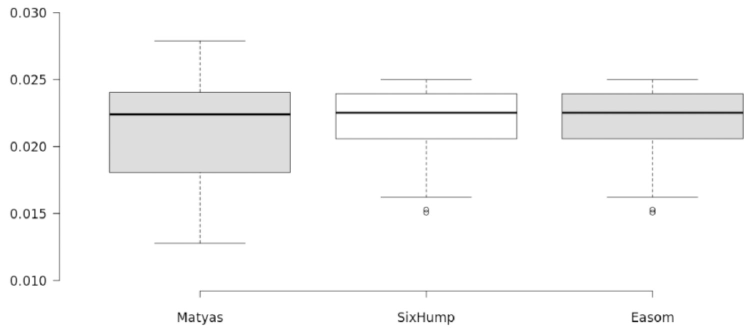

Fig. 6 *Diversity boxplot*

3.4 Advantages and limitations

The problem of computing high-level representations of point clouds lies in the manipulation of such amount of information. In our case, the process is highly dependent on the number of PPU used to represent each surface or polysurface and the number of individuals shown to the user. In case of sampling PPU, the computational complexity of the genetic algorithm is $O(n \log(n))$, noting that performing point cloud with sizes over 10,000 points in home computers is untreatable. However, the complexity attending the population size is $O(\log(n))$. Hence, the algorithm complexity is $O(n \log(n))$.

3.5 User influence

Ten (six men and four women, age range 18–26 years) participants from the University of A Coruña with formal artistic training took part in the study. Participants were asked to evolve a sphere until they judged to be pleasant (as shown in Fig. 2). Fig. 7 shows the heatmap related to evolved spheres by means of the R^2 correlation value. The average R^2 value between final designs was 0.1327 ± 0.0134. Results show design's low correlation between participants. This seems to indicate great relevance in accordance of users influences on the output through their choices during the evolutionary process.

	1	2	3	4	5	6	7	8	9	10
1	0.1347	0.1281	0.1332	0.1239	0.1238	0.1665	0.1228	0.1248	0.1360	0.1248
2	0.1249	0.1259	0.1634	0.1312	0.1253	0.1720	0.1339	0.1253	0.1274	0.1238
3	0.1237	0.1278	0.1296	0.1258	0.1692	0.1313	0.1315	0.1258	0.1311	0.1242
4	0.1280	0.1284	0.1235	0.1542	0.1274	0.1267	0.1256	0.1253	0.1256	0.1267
5	0.1241	0.1246	0.1270	0.1270	0.1241	0.1411	0.1658	0.1284	0.1346	0.1249
6	0.1237	0.1382	0.1413	0.1317	0.1254	0.1307	0.1430	0.1464	0.1310	0.1248
7	0.1247	0.1250	0.1338	0.1254	0.1250	0.1265	0.1339	0.1342	0.1241	0.1656
8	0.1372	0.1365	0.1387	0.1248	0.1248	0.1392	0.1597	0.1234	0.1236	0.1285
9	0.1227	0.1302	0.1259	0.1256	0.1231	0.1323	0.1276	0.1265	0.1424	0.1319
10	0.1239	0.1296	0.1250	0.1328	0.1267	0.1267	0.1740	0.1928	0.1258	0.1241

Fig. 7 *User influence heatmap in terms of* R^2

4 Application examples

The algorithm has been tested for the design and fabrication of different 3D objects, with applications for industrial design, architecture, sculpture and replication of natural structures, among other possibilities. A simple example of the application of the proposed evolutionary morphogenesis is the production of an industrially designed object to be later mass produced, a cast ceramic tile with an irregular surface. By applying the GP approach to a flat surface, families of 3D tiles have achieved, sharing similar irregularities but all geometrically different (Fig. 8). The same algorithm can also be used for the auto-generation of 3D irregularities or deformations on curved surfaces (Fig. 9).

Fig. 8 *Example of an irregular 3D tile design*

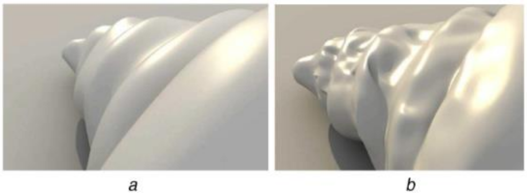

Fig. 9 *On the left smooth curved 3D geometry. On the right evolutionary morphogenesis generation of irregularities through the proposed GP technique*

In the case of the presented system, elements as distribution of forces or cost are not taken into account. In any case, any constraint could be easily implemented as fitness functions, even applying several simultaneously. For example, many aspects of the cost of building a structure can be computed as numeric values which could be used to design ‘cheap’ design instead of ‘expensive’.

The novelty about this process is that what drove the result has been a point-based algorithm by a parametric and evolutionary process, and in particular, the genetic selection of successful designs in order to naturally leads to a final design result. The designer instead of drawing in 3D will run the system by modifying the parameters, to later select and recombine the successful geometries following an evolutionary process. Other industrial design applications are the production of sculptural designs, to be later fabricated by using 3D printing (by directly 3D printing or by making 3D printed cast melds). Some examples are irregular ceramics, sculptures or jewellery, just to name a few (Fig. 10).

Fig. 10 *Examples of irregular surfacing designs (ceramics, sculptures and jewellery) that could be easily generated and fabricated by the use of the proposed GP morphogenesis techniques*

A larger-scale application example is the design a complex curvature facade for a skyscraper. As reference for the geometrical exploration two built examples will be used, the Beekman Tower designed by Frank Gehry in New York City and the Aqua Tower in Chicago by Studio Gang (Fig. 11).

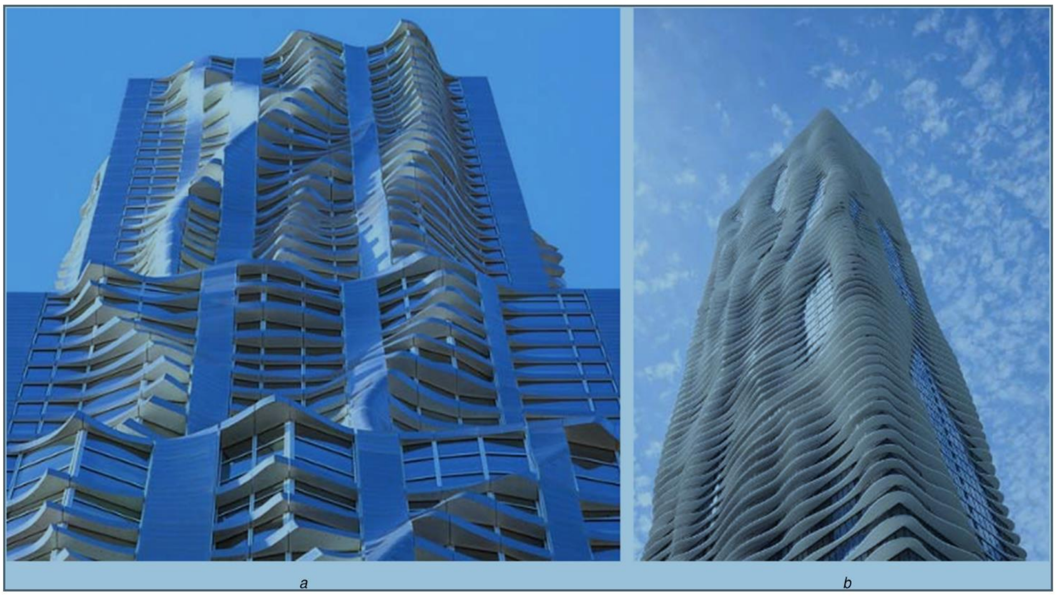

Fig. 11 *On the right, the Beekman Tower, by Frank Gehry, completed in 2010. On the left, the Aqua Tower, by Studio Gang, completed in 2009*

The Beekman Tower has an exterior envelope made as a continue morphing surface of stainless steel panels, and will be used as a start point reference for the design. Per Sundberg [27] parametric modelling in SolidWorks was used for the design and CATIA for the fabrication of the complex undulating shapes. This time instead of using parametric modelling a GP approach has been tested for the generation of similar type of design, by applying an evolutionary point-based methodological process.

The pseudocode has been applied to a flat surface to generate an experimental skyscraper facade design. For such a purpose, the deformation parameters have been increased, until desirable effects were achieved. The algorithm has assisted not only to produce the deformations itself by modifying the parameters but also into a selective design process, by producing more design options based on the user design deformations picks, eventually generating hundreds of families of design results, leaving to the user the final decision to pick one of them to be later implemented as the final design choice. The parametric process and further genetic evolution of successful geometries has driven the process. One of the hundreds of solutions given by the algorithm has been selected, and then to produce a final design of a building a simple pattern of windows was projected on it. On purpose, deformations that will produce a similar design than Gehry's Beekman Tower and Gang's Aqua Tower were selected, but this time a combination of parametric design and evolutionary selection has driven the process (Fig. 12).

Fig. 12 *3D model of tower design using GP morphogenesis*

Another possible application is the 3D replication of natural systems for 3D rendering or animations. As the proposed GP is a point-based algorithm, it is suitable for the replication of natural systems made of particles. Some possible examples are the generation by evolutionary computation of dunes or water geometries (Fig. 13). The examples described are just some of the multiple possibilities of the proposed interactive evolutionary morphogenesis process.

Fig. 13 *3D model of water generated through evolutionary morphogenesis modelling*

5 Conclusion

The purpose of the research was to modify surfaces by evolutionary morphogenesis processes. The development of a point-based genetic morphogenesis algorithm has been able to generate diversity by producing irregularity on a surface; furthermore, it has been achieved through an evolutionary process. To further test the approach, it has been successfully used for the generation of new industrial, sculptural and architectural designs, in that sense the morphogenesis approach has proven itself successfully into the generation of morphologies through evolutionary selection to be later implemented as final designs. It has also been successfully tested for the replication of natural particle systems, such as dynamic water surfaces.

The unique characteristic of our process of modelling is user's ability to modify and evolve the point cloud by a genetic recombination of surviving samples, not by manually manipulating the parameters. The use of point clouds seems ideal for a design methodology based on bottom-up self-organising processes, as the genetic algorithm proposed. This is due to the fact that these systems are made of unconnected simple elements, creating in this case complex surfaces. The use of point-based evolutionary morphogenesis has resulted successful into redesigning the geometry of complex irregular surfaces. The use of point clouds seems ideal for a design methodology based on bottom-up self-organising processes, as the genetic algorithm proposed. This is due to the fact that these systems are made of unconnected simple elements, with simple rules of interaction, creating in this case complex surfaces.

6 Acknowledgment

This work was supported by the General Directorate of Culture, Education and University Management of Xunta de Galicia (Ref. GRC2014/049).

7 References

[1] Vasari, G., Foster, J.: 'Lives of the most eminent painters, sculptors, and architects' (George Bell, London, 1872), v. 3.
[2] Dürer, A., Strauss, W.: 'The painter's manual: a manual of measurement of lines, areas, and solids by means of compass and ruler assembled: for the use of all lovers of art with appropriate illustrations arranged to be printed in the year MDXXV. The literary remains of Albrecht Dürer' (Abaris Books, New York, USA, 1977), pp. 1–472.
[3] Monge, G., Zalt, K., Halbleder-Einband, R.: 'Géométrie descriptive: leçons données aux écoles normales, L'An 3 de la république' (Baudouin, Imprimeur du Corps législatif et de l'Institut national, Thiverval-Grignon, France, 1798).
[4] Rocker: 'Interface: between analog and digital systems' (2010).
[5] Durand, J.: 'Précis des Leçons d'Architecture: données a l'école royale polytechnique' (No. v. 2 in Précis des Leçons d'Architecture: données a l'école royale polytechnique. Chez l'auteur a l'École Poytechnique, 1817.
[6] Dunn, N.: 'Digital fabrication in architecture' (Laurence King Publishing, Durrington, UK, 2012).
[7] Marcos, C.L.: 'Complexity, digital consciousness and open form: a new design paradigm' (2010).
[8] Turing, A.: 'The chemical basis of morphogenesis', Philos. Trans. R. Soc. B, 1952, 237, pp. 37–72.
[9] Johnson, S.: 'Emergence: the connected lives of ants, brains, cities, and software' (Scribner, New York, USA, 2012).
[10] Leach, N.: 'Digital morphogenesis' (2009).
[11] Delanda, M.: 'Deleuze and the use of the genetic algorithm in architecture' (2005).

[12] Andersen, P., Salomon, D.: 'The architecture of patterns' (W. W. Norton, New York, USA, 2010).
[13] Jiang, C., Tang, C., Tomicic, M., et al: 'Interactive modeling of architectural freeform structures - combining geometry with fabrication and statics', Advances Architect. Geom., 2014, 2014, pp. 95–108.
[14] Tang, C., Sun, X., Gomes, A., et al: 'Form-finding with polyhedral meshes made simple', ACM Trans. Graph., 2014, pp. 70–79, 33–34.
[15] Fraternali, F.: 'A thrust network approach to the equilibrium problem of unreinforced masonry vaults via polyhedral stress functions', Mech. Res. Commun., 2010, 37, (2), pp. 198–204.
[16] Holland, J.: 'Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence' (University of Michigan Press, Ann Arbor, USA, 1975).
[17] Goldberg, D.: 'Genetic algorithms in search, optimization, and machine learning' (Artificial Intelligence, Addison-Wesley, Boston, MA, USA, 1989).
[18] Bentley, P.: 'Evolutionary design by computers. No. Part 1' (Morgan Kaufman Publishers, Burlington, USA, 1999).
[19] Renner, G., Ekárt, A: 'Genetic algorithms in computer aided design', Computer-Aided Design, 2003, 35, (8), pp. 709–726.
[20] Dawkins, R.: 'The blind watchmaker' (Penguin Science. Penguin Books, London, 1991).
[21] Todd, S., Latham, W.: 'Evolutionary art and computers' (Academic Press, Orlando, USA, 1992).
[22] Rovira, J., Wonka, P., Sbert, M., et al: 'Point sampling with uniformly distributed lines'. Point-Based Graphics, Stony Brook, USA, 2005, pp. 109–118.
[23] Machado, P., Cardoso, A.: 'NEvAr – the assessment of an evolutionary art tool'. AISB'00 Symp. Creative and Cultural Aspects and Applications of AI and Cognitive Science, Birmingham, UK, 2000.
[24] Machado, P., Cardoso, A.: 'All the truth about NEvAr', Appl. Intell., Special Issue Creative Syst., 2002, 16, (2), pp. 101–119.
[25] De Jong, K.A.: 'An analysis of the behavior of a class of genetic adaptive systems'. PhD thesis, Ann Arbor, MI, USA, 1975, AAI7609381.
[26] Molga, M., Smutnicki, C.: 'Test functions for optimization needs' 2005.
[27] Sundberg, D.: '8 spruce street. Metals in construction' (The Steel Institute of New York, New York, USA, 2011).