IET Image Processing

Case Study

A point-based redesign algorithm for designing geometrically complex surfaces. A case study: Miralles's croissant paradox

ISSN 1751-9659
Received on 13th February 2020
Revised 23rd April 2020
Accepted on 11th May 2020
E-First on 3rd September 2020
doi: 10.1049/iet-ipr.2020.0223
www.ietdl.org

Adrian Carballal[1,2] ✉, Rafael Iván Pazos-Pérez[1], Nereida Rodriguez-Fernandez[2], Iria Santos[2], María D. García-Vidaurrázaga[3], Juan R. Rabuñal[4]

[1]Department of Computer Science and Information Technologies, Faculty of Computer Science, University of A Coruña, A Coruña, Spain
[2]CITIC-Research Center of Information and Communication Technologies, University of A Coruña, A Coruña, Spain
[3]Department of Construction Technology, University College of Technical Architecture, University of A Coruña, A Coruña, Spain
[4]Centre of Technological Innovation in Construction and Civil Engineering (CITEEC), University of A Coruña, A Coruña, Spain
✉ E-mail: adrian.carballal@udc.es

Abstract: This study explores the use of point clouds for both representation and genetic morphogenesis of complex geometry. The accurate representation of existing objects of complex curved geometry, which are subsequently geometrically modified by evolutionary morphogenetic processes, is analysed. To this end, as a method of representation and generation of complex geometries, a point-based genetic algorithm and the use of large unstructured point clouds are proposed. A study of convergence and diversity of the implemented algorithm is detailed, as well as a comparison with the Coyote optimisation algorithm in terms of representation error demonstrating its efficiency. Some commonly used three-dimensional formats in architecture, such as NURBS and polygon meshes, are analysed, and compared against point clouds. This study also includes an evaluation regarding whether the use of point clouds is a more suitable format for realistic representation, rationalisation and genetic morphogenesis.

1 Introduction

Non-orthogonal curved geometry has always been present in architecture to some degree. Complex geometric elements were typically built on the basis of traditions that replicated or improved the experiences of past times and geographical areas. Many of this intricate geometry was for decorative purposes only. Some curved geometry elements sometimes have structural purposes, e.g. Gothic arches and vaults, or domes. It has its origins in Renaissance architectural representation. Leone Battista Alberti, a precursor to contemporary visualisation, figures out the basic rules of linear perspective as specified by [1] in his book, 'Lives of the most eminent painters, sculptors and architects'. The findings of Alberti on architectural perspective and the further discoveries of Dúrer and Strauss [2] provided architects with dramatically complex representational tools for the first time. New experiments started to be carried out, departing from previous knowledge passed from generation to generation, by returning to and reinterpreting the classical architecture. In particular, Andrea Palladio began to depart from the old classical style and started a process of experimentation that eventually culminated with Francesco Borromini, beginning a new era of complex formal architecture, known today as Baroque. The architecture of San Carlo Alle Quattro Fontane, by Borromini, completed in 1646, was a direct consequence of the two centuries before that architectural image. According to [3], though visually complex, the geometry of Borromini followed well-known geometric forms, such as ovals, triangles and ellipses. Having finished the formal and ornamental excesses from the Baroque Era, classical rigor took over again, during the Age of Enlightenment, in post-revolutionary France, called Neoclassicism. At this stage, descriptive geometry was developed, and drawings were standardised. As described by [4], the mathematician radicalises geometry by constructing an orthogonally based system for representing complex geometry, incorporates mathematical calculations and visualisation in the form of complex curved trajectories, and provides the basis for contemporary representation of complex surfaces and orthodontics [5].

The invention of descriptive geometry provided architects with representational tools to precisely control and imagine even more complex geometric forms, whether regular or not, opening up new possibilities for exploring geometric complexity. Designs as a codified form of representation were not developed until the beginning of the 19th century, when the Monge [5] findings were used to create a orthogonal system based on a Cartesian map [4]. During the 19th century, the evolution of representation led to the creation of a wide variety of architectural styles that reached their peak with the movements of Art Nouveau, Modernism and Art Deco, all highly formal styles with complex curves, like Gaudi's work, among many others. Regardless of the incorporation of modern materials and technology into architecture throughout the 20th century, architecture reinvents itself in a revolutionary break from the formalistic past, sparking the modern movement. It is not until the late 1990s that due to the introduction of computers modern architecture experiences a fresh and rapid transformation. The representation and construction of buildings with complex geometry became a particularly important topic, as computation allowed not only digital modelling, but also digital fabrication of geometries, difficult to manage until then. By the late 1990s, architects also began to focus on other software packages that were originally designed for three-dimensional (3D) graphics, particularly for the gaming, animation and motion picture industries. Such applications were 3D-oriented and targeted at simulating reality. In the beginning of the 21st century, not just the use of some of these 3D software packages were popular in the architecture industry, but the introduction of software packages specifically designed for architecture began to incorporate these 3D functions to allow architectural representation to leave Euclidean orthogonal geometry [6] with the main aim of transforming them directly into 3D. Throughout recent years, architecture has seen a growing number of buildings as a direct result of the invention of computers, which have adopted extremely complex curved geometry not previously seen. The construction of Frank Gehry's Guggenheim Bilbao Museum was quickly followed by a series of asymmetry, anomalies and non-recurring forms in worldwide geometrically complex structures, which took up this technique in 1997.

Computers have not only made it impossible beforehand to control and directly produce shapes and structures but have also opened up new avenues of experimenting in architecture. More recently, architects have started to take an interest in the idea of using digital tools to create their own designs. Computers have not only made it impossible beforehand to control and directly produce shapes and structures, but have also opened up new avenues of experimenting in architecture. In recent times, architects have begun to take an interest in making designs themselves using computational instruments.

Nowadays methods included generative design using algorithms, leveraging computer analytical ability to overcome the inherent human limitations [7]. The aim is to describe structural structures generated through a script using generative algorithms. Turing [8] used the word morphogenesis in his 1952 article titled 'The Chemical Base of Morphogenesis', where he discussed the repetitive numerical patterns of flowers, showing how flowers developed their complex geometry through self-organising processes. Turing, as a WWII code breaker and also one of the fathers of modern computing, focused on pattern recognition in an apparent ambiguity. The idea was clear, complex patterns can come from very simple rules, creating complex self-organised structures through the local interactions of simple unconnected components, a process also called the emergence [9]. Morphogenesis focuses primarily on the bottom-up philosophy of shape discovery, stressing the success over the existence [10]. Open forms should be defined as topology rather than geometrical, as noted in [11], since it is the relationship of the sections of the forms that determines the whole. Closed shapes are geometrically defined but open shapes do not necessarily fit into geometrical space. The genetic processes, the replication and the evolutionary elimination [12] can also contribute to morphogensis. Biological development is a selective mechanism comprising variability drivers such as spontaneous changes or synaptic changes.

All these systems are based on topological systems such as point clouds as they are similar to auto-organised structure and bottom-up logic, such as the organism's cells and the biomachine. They are more closely connected. Point-cloud use will be addressed in greater depth both for the depiction of complex geometry and for genetic morphology.

Metaheuristics are currently used to solve problems that were considered intractable. There are many known types of metaheuristics, from the most classical to the new generation. So far, genetic algorithms (GAs) are the most widely used with at least 1270 reported studies using them [13]. The technique was developed in the 1970s by John Holland and his students at the University of Michigan for studies of adaptive systems [14]. The first publication on the use of GA applying natural inspirations was made by Goldberg in 1989 [15]. Since then until now, various studies have been carried out in this regard. A complete reference for the latest generation metaheuristic algorithms with natural inspiration can be seen in Yang's 'Nature-inspired metaheuristic algorithms' (2010) [16]. There are a number of more current publications that also cover metaheuristics and its applications as 'A survey on optimisation metaheuristics' Boussaïd *et al.*, 2013 [14], 'Metaheuristics in large-scale global continues optimisation: A survey' de Mahdavi *et al.*, 2015 [17] y 'A survey on new generation metaheuristic algorithms' de Dokeroglu *et al.*, 2019 [13]. Thanks to them and the studies they address we can know that GA can implement in different ways depending on the problem and its use can be made in different areas (robotics, software engineering, optics and image processing, etc.) [14].

The most current known applications on naturally inspired metaheuristics have been carried out between 2018 and 2020. Shayanfar and Gharehchopogh, 2018 [18] create farmland fertility, an algorithm that divides agricultural land into different sections in terms of soil quality based on the fertiliser used by farmers with the aim of improving fertiliser use. Pierezan and dos Santos Coelho, 2018 [19] propose a metaheuristic for global optimisation called Coyote optimisation algorithm (COA). COA is an algorithm inspired by the species of *canis latrans* that inhabit North America. COA considers the social organisation of coyotes and their adaptation to the environment. The same authors have made progress in applying COA to other problems, as can be seen, e.g. in their article 'Cultural COA applied to a heavy duty gas turbine operation', 2019 [20]. On the other hand, Klein *et al.*, 2018 [21] propose an optimisation algorithm based on the cheetah (CBA), capturing the social behaviour of these animals. This CBA algorithm was validated against seven known optimisers using three different reference problems. Also in 2018, the same authors presented a meerkat-inspired algorithm (MEA) [22], a population-based swarm intelligence algorithm for global optimisation in the continuous domain. In 2019, Yapici and Cetinkaya [23] proposed a new metaheuristic algorithm inspired by the collective movement of groups of animals that mimicked the leadership hierarchy of swarms to find the best feeding area or prey. The proposed method was tested on some optimisation problems to show and confirm performance on test stands. This algorithm is able to converge the global optimum and avoid the local optimum effectively. In addition, it is designed for multiple target problems (MOPFA). Finally, de Vasconcelos Segundo *et al.*, 2019 [20, 24], carried out two new metaheuristics: owl optimisation algorithm, which is inspired by the behaviour of decoy that owls exhibit when they detect threats close to their nests; and Falcon optimisation algorithm based on the hunting behaviour of falcons. Another example with a different basis is that of Moosavi and Bardsiri [25], which presents an algorithm inspired by the efforts of the two groups of poor and rich to achieve wealth and improve their economic situation. Finally, one of the most recent works in the field of metaheuristics is that of Sulaiman *et al.*, 2020 [26], which presents an algorithm that mimics barnacle mating behaviour in nature to solve optimisation problems.

2 Representation of complex geometry: Miralles's croissant paradox

One example of precomputer representation of complex geometries is an essay by architect Enric Miralles [27], titled 'How to lay out a croissant'. It was indeed an extremely difficult task at the time to draw a croissant, since its geometry consists of irregular and complicated curved surfaces arising from a baking process like matches, but not exactly, all croissants.

The cartesian precision of his plans and parts and in particular their proportions was not as clear a process and many inaccuracies were studied. Miralle [27] suggested measuring the plane over a number of triangles (which are non-defortable geometric models), then drawing the sections, focussing primarily on straight tangent parts, then specifying the curvature radius and alloting the centres thereof. (Fig. 1). By properly describing the curvature radius and tangents, [27] rationalised croissant geometry, transforming into lines and circles a set of continuous irregular surfaces.

This sought to transform a highly irregular geometry into an effective geometry. Nonetheless, it is very clear that the croissant drawn by him was inherently different, although its geometry was very similar to the original, as there was not enough rationalisation to explain the imperfections of the original geometry.

3 Digital modelling and visualisation of complex geometry: polygon meshes and NURBS

The croissant example is used for further explanation of Miralles' geometric and representative problems, for simple comparisons with pre-computer and post-computer presentations; however, any other curved shape could also be used. Various 3D formats and applications are used for our research to model and simulate a croissant, so that the results are analysed subsequently. Furthermore, certain of the most widely used software programs for architectural design and representation can be used to handle comprehensive curved geometry such as AutoCAD, Rhinoceros and 3DSMax. To this end, a croissant was first sliced and measured manually (Fig. 2), generating plans and parts as accurately as possible, and finally using CAD software the 2D orthogonal projections were drawn (AutoCAD).

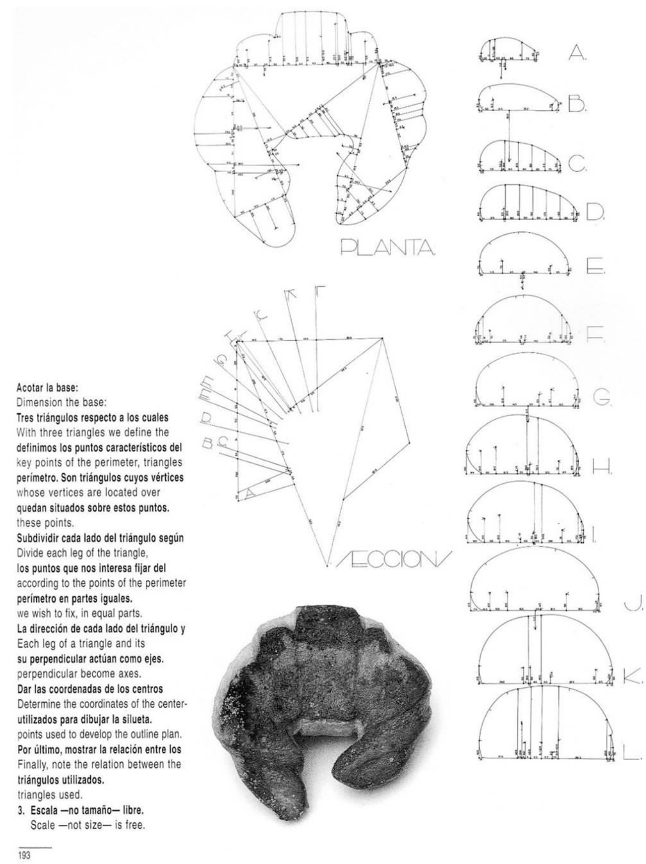

Fig. 1 *Miralles and Prats's article 'How to lay out a croissan'*

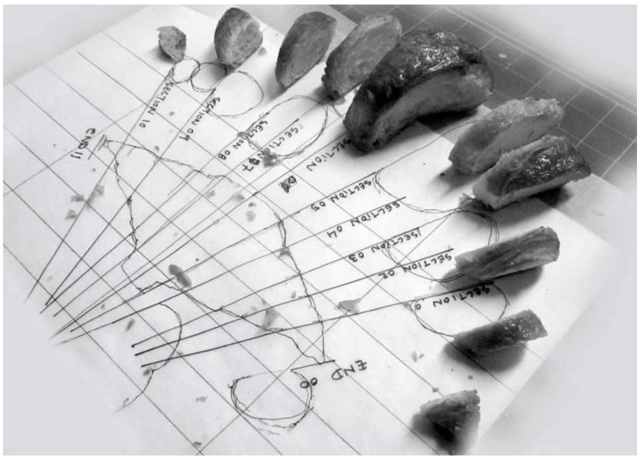

Fig. 2 *Manual slicing and measurement process*

Next, CAD sections and drawings were shipped to Rhinoceros 5.0 and based on NURBS because the programme's curved shape is very precise, similar to that of a croissant surfacing. They designed the splines of the NURBS to create a smooth continuous coating. The resultant geometry was convincing due to the loft approach from the original parts, but all the inconsistencies and imperfections on a true croissant were overlooked. The croissant geometry has been manually designed in order to achieve a more accurate and similar result with its initial geometry (Fig. 3).

The use of smooth after NURBS bonding makes the result optimal, but can be improved through manual modelling. With such manual modelling the detail increases and the number of NURBS is multiplied. This makes the definition more precise and the inconsistencies and imperfections missed in the initial model disappear. In Figure you can see the result of the direct modelling from the sections (images at the top) and the manipulated manual modelling with which the result is improved (images at the bottom).

The model was then converted into a polygon grid and then brought into Autodesk 3DSMax to create a realistic representation of the croissant (Fig. 4), using the VRay rendering engine developed by Chaos group [28]. The integration of the two software packages provides some of the most accurate design rendering tests (Fig. 4). In addition, a jpg map was added to the diffuse, texture and bump channels using UVW cylindrical mapping (Fig. 5) in order to obtain an even more unusual surface and natural colour and texture, from images taken from the original croissant.

Although very similar, the croissants differed greatly from the original, whether in geometry or visualisation, since the new geometry reflects a 3D manual solution to the geometries of the original croissant.The projection is of the initial texture of the croissant, but the picture pixels do not fall precisely at the same sample spot. In addition, all the croissants shown in Fig. 5 are the same croissant. Since no croissant is identical with another in the real world, it would only be possible to create variety by manually designing one by one differently, or by manually making modifications on each of the candidate designs' surfaces. The results obtained were of a high degree of accuracy by using Polygon mesh and NURBS geometry.

The geometry created, though very similar, still differs from the original sample over the whole surface in $+/-2$ mm. Furthermore, the croissants lack the inherent imperfections of the original, made manually, and in particular the consistency of a batch of real croissants. All the clones would be identical at this point. To further solve the representational problem, two new additional steps are proposed to correct some of the deficiencies found:

- To get more accurate measurements, it is suggested to use a LiDAR scanner, creating a replica of the croissant's original geometry as a point cloud format, which is a technique not widely used in architecture.
- In order to achieve the surface imperfections and diversity, a morphogenesis process regulated by a GA is also suggested.

4 Laser scanning and point-based visualisation of complex curved geometry

LiDAR 3D scanning technology has been used to scan another croissant. Laser scanners use massive point clouds to generate data, and the data from scans are commonly called unstructured massive points clouds, which are basically extensive unconnected point databases without a specific structure or order. Usually these data sets contain their levels of co-ordinates, colour and luminosity. The scans range from tens of thousands of points to millions.

Initially proposed by [29], the point-based rendering (PBR) concept suggested an approach based on working directly with point clouds, rather than polygon meshes or NURBS. The concept for QSplat was implemented first with the [30], a multi-resolution hierarchical sphere /PBR framework, and to structure data using rendering algorithms, which describe explicitly visibility violation, information level and rendering. Instead, the points would be rendered as splats, using triangles, circles or Gauussians. PBR makes it possible to monitor millions of points since the manageability and effectiveness of points in processing and rendering is much better than in polygon-based systems, which is particularly true for large models or complex curved geometry scans (Fig. 6).

To this end, our research group has created an OpenGL PBR software package named ToView, which is used in real time. ToView is able to handle point clouds of billions of points easily, which standard PCs can do with a great degree of precision and realism in real time.

Depending on the desired speed and result the program will work with three different visualisation algorithms. The first algorithm of visualisation uses the image-aligned squares to make points. This algorithm has the highest performance; splats are made on each splat using just one OpenGL level. The next algorithm in the framework uses affinally projected point sprites, this algorithm yields a better representation of the splat form than the one seen in [31]. A shader can delete the part of the square outside the splat for

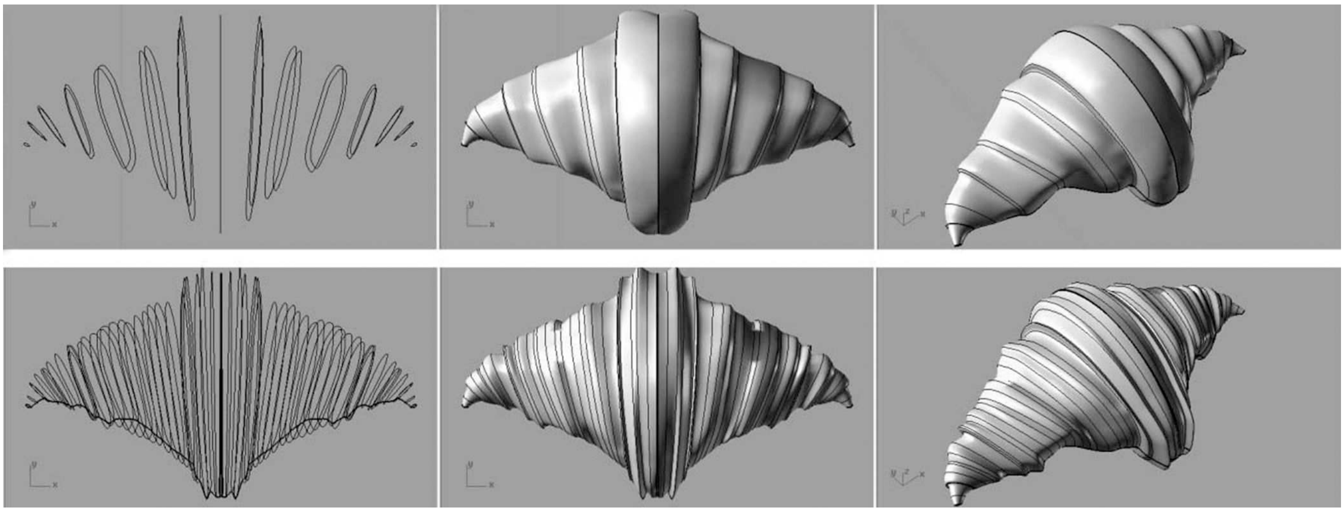

Fig. 3 *Process of modelling the croissant automatically through software and with subsequent human manipulation. The top row shows the result of manually measured croissant sections being modelled directly from the ones. The bottom row shows how the surfaces are manually manipulated to achieve a more convincing representation*

Fig. 4 *3D croissants renderings. The picture on the left shows the croissant model designed in 3D. The illustration to the right shows the croissant's surface with additional manual texturing*

Fig. 5 *3DSMAX and VRAY croissants, adding jpg diffuse, bump mapping and reflection*

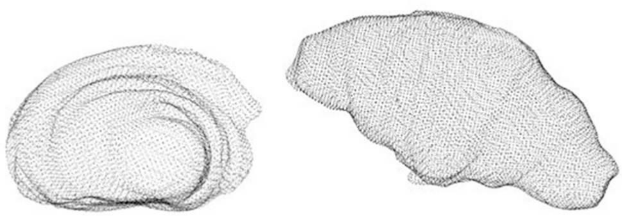

Fig. 6 *Unstructured point cloud geometry, resulted from the LiDAR laser scan*

each of the squares.This approach is still troublesome when the splatting is in drastic angles leading to surface holes. The last algorithm in the developed software is the right point rasterisation. The technique present in [32] correctly transforms the outer splat contour, but encounters difficulties with the splat centre. [31] implemented a more powerful and perspective algorithm, which was modified to fit the needs of the system. This algorithm has been originally designed to represent ellipses to handle the oriented discs used in our case study (Fig. 7).

ToView is also capable of making dots of different sizes for cases of abnormal point cloud density. This is important for reducing clipping errors and improving the quality of the rendering. The point sizes are obtained in a pre-processing stage, in which it is possible to estimate certain point features required for advanced point rendering. In addition, the software also supports MSAA anti-aliasing [33] to further boost the rendering efficiency. This technique takes away aliasing artifacts and improves the quality of the surface. LiDAR scanning and PBR allow for a more precise 3D simulation of the croissant in real time, with a precision level of up to half a millimeter, so that it can be claimed that the original problem was partially solved (Fig. 8). The surface however has natural imperfections that can not be rendered by manual measurement and modelling.

Laser scanning has its own disadvantages, as it is difficult to scan other areas, with the laser beam (shadow) concealing its shape, and other minor inaccuracies.

The rendering results are acceptable based on point-cloud representation, with visual quality comparable to that provided by NURBS manual modelling and VRay rendering (Fig. 9). A much similar result was obtained from the original sample using point-based scanning and rendering. Point geometry and description in this way are more precise than meshes.

The final visual effect is not the focus of our evaluation process; we are more interested in the geometry itself, together with the possibility of further manipulations of the model.

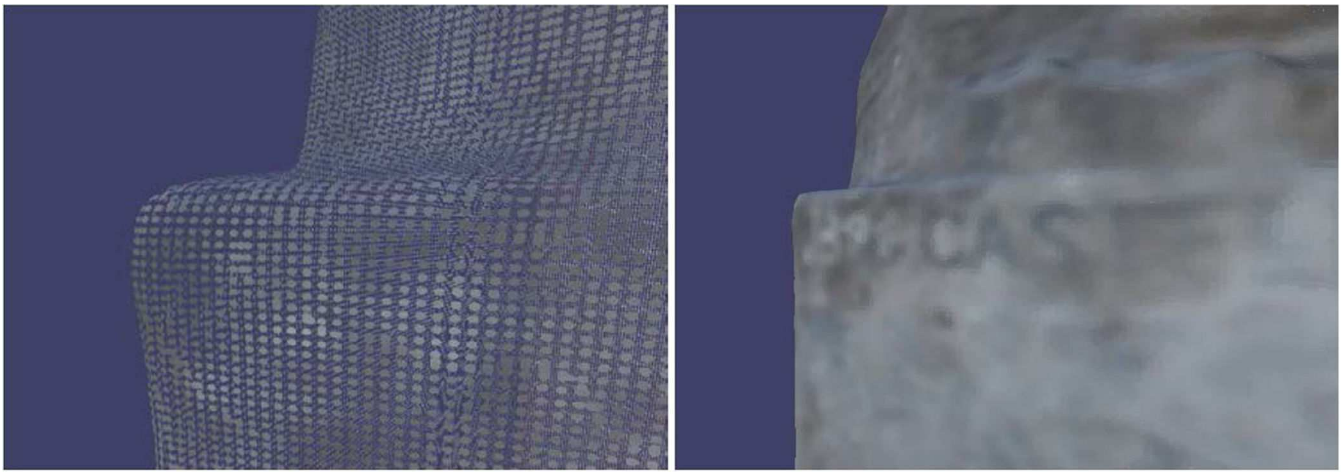

Fig. 7 *ToView perspective correct rasterisation algorithm, using PBR of large unstructured point clouds with aligned disk*

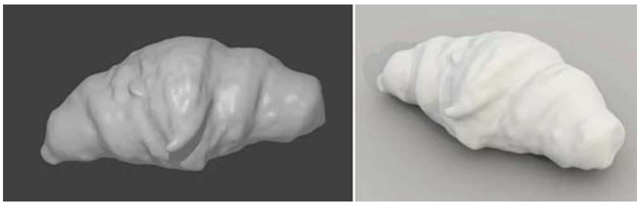

Fig. 8 *PBR from a LiDAR scan of a croissant with baked illumination of large unstructured point cloud. On the left-hand side-a point cloud rendering in ToView, on the right-hand side-its conversion into a polygon mesh and VRay*

Fig. 9 *On the left: Point-Cloud mesh centered upon a huge unstructured point cloud from a croissant LiDAR scan. To the right: the polygon meshes manually measured and modelled rendered with VRAY in 3DSMax*

5 Genetic-based morphogenesis approach

Further geometric manipulation of the original object is also of interest to us. It also explores the generation of deformations on the croissant's surface, and particularly the ability to generate diversity, by creating families that result from manipulating the original form. To this end, a process is proposed for the topological evolutionary morphogenesis. A GA was developed to incorporate the imperfections on the surface of the croissant and create a number of similar solutions. The aim is to create inaccuracies on the surface, not manually but by means of a morphogenesis method, from the point-based geometry of an existing croissant as a start. Furthermore, a wide range of similar samples without a comparable solution of the original 3D geometry are automatically generated.

This study proposes an GA [34, 35] that modifies a surface appearance by altering its representation as a point-cloud, with a goal of obtaining new designs. Evolution is probably one of the best discovery methods to date and has for centuries been applied with a clear interest in nature, so it seems much more than reasonable to extend it to the field of creative design and today it has a widespread presence [35] Bentley:1999.

The breakthrough is that genetic code is used to render a natural texture of the point cloud on the original surface. Not the user, but the algorithm, determines how the surface is deformed, like rhinoceros, grasshopper, or any other parameter packaging. Instead, the user determines which of the programs examples conforms to his requirements just like the users' definition [36, 37].

Such human–machine interaction has certain benefits in particular in the creative process. The user chooses the requirements that they deem appropriate in the implementation process without ‘wasting’ resources and tend to be more time-consuming.

The algorithm is based on interactive evolutionary computation (IEC), which mimics the evolutionary selection/reproduction process. Selection would ensure the most suitable live for reproduction, thereby ensuring the succession of the most suitable genetic material from the descent. Identifying better individuals within the space of all possible individuals [38] may also be called a method of search. The user assesses the individuals that lead the evolutionary process. The success of this strategy depends on the user's willingness to use any or all of the different criteria (objective or subjective) to classify the individuals identified as possible solutions.

5.1 Surface redesign algorithm based on genetic morphogenesis deformations

The genetically engineered model simulates natural selection mechanisms: fitness survival (selection) recombined with minor changes (mutation) in conjunction with the most suitable genetic material (crossover). The algorithm's aim is to enhance the design experience, so designers are provided with various solutions for adoption as their own or to assist them in the creative process. It's not a matter of supplying them with a tool to ‘transform’ the creative work of the artist, but rather of using it as a creative tool in which the user retains control. The design-cycle process consists of the following steps:

(i) The algorithm generates a random initial population.
(ii) The user deletes all those which do not match criteria.
(iii) The program breeds population phenotypes through the recombination and mutation of the genetic codes (genotypes) of the current population, preserving undeleted solutions.
(iv) Return to point (ii) until the user finds a consistent solution.

Genotype is a mathematical expression in tree-shaped form in our system. The trees are built as tree nodes and terminals as leaves out of a lexicon of binary functions. The terminal collection in the Cartesian model consists of the variables ‘x’ and ‘y’. This definition of morphogenesis deformation is based on NEvAr, developed by [39] but extended to the design of 3D surfaces rather than the design of the images. In the case of NEvAr a perception of genotype (an individual) results in an image-like phenotype. We evaluate the expression for each pixel coordinate to produce an image, and the output is interpreted as the grayscale value of the corresponding pixel [40]. In comparison, our first algorithm tests the point-cloud with a fixed density that represents the initial geometry. Once the arithmetic expression is evaluated, a few modifications are made directly in the point cloud, so that the value extracted decides its displacement above normal at every point (see

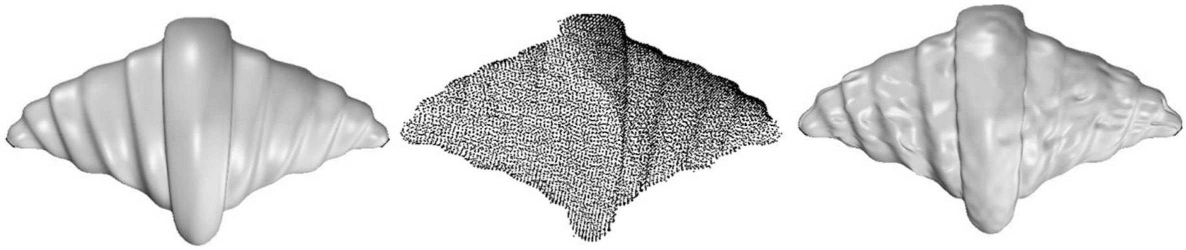

Fig. 10 *Transformation of the Croissant representation that occurs during application of the algorithm. From left to right: geometry made, changed point-cloud and rendered again*

Table 1 The algorithm predefined parameters used for the design process

Parameter	Value
point-cloud UV density	50 by 50 pps
range of displacements	[-0.1; 0.1] cm
minimum percentage of affected surface	95%
maximum Tree Size	40
minimum Tree Size	15
crossover rate	85%
mutation rate	15%
functions (Tree Nodes)	+, –, *, /, %
terminals (Tree Leaves)	‘*x*’, ‘*y*’

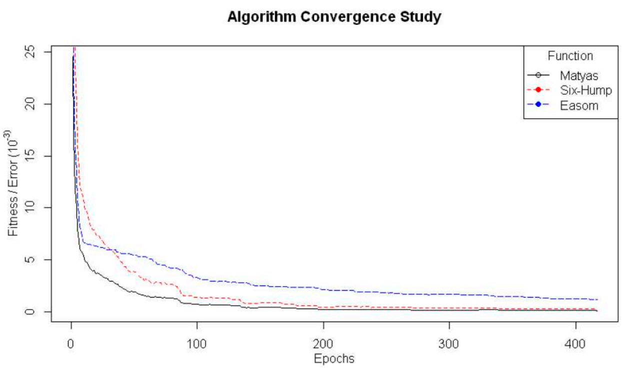

Fig. 11 *Convergence graph for the three purpose functions*

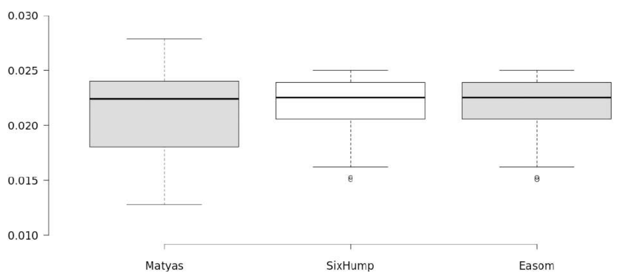

Fig. 12 *Diversity boxplot*

[41] for further details). The resulting new geometry is assembled when complete (Fig. 10)).

The number of parameters initially preset defines how the deformations are implemented, such as point-cloud distance, range of min/max displacements, minimum percentage of the affected surface, maximum and minimum size of the genotype tree's first population and those related to the evolution process (mutation and crossover). Table 1 defines all of those parameters used in the case study below.

We will now give a brief explanation of each of the parameters in Table 1. The parameter ‘Point-cloud UV density’ specifies the number of points per surface that is part of the figure. These points will be equidistantly positioned along the surface. Range of displacement indicates the range in which each point of representation can move along the normal. Minimum percentage of affected surface indicates the minimum percentage of points representing a surface that must change its position. Maximum and Minimum Tree size indicate the maximum and minimum size, respectively, of the tree representation of the formula that determines the modification of each point. Any candidate solution that does not satisfy this range will not be taken into account in the search process. Functions (tree nodes) are the mathematical formulas that will be in the non-terminal nodes of the tree representation and the terminals may only be the values corresponding to the X and Y axes.

Roulette wheel partner selection utilises a basic canonical GA [42]. For recombination, a basic GP crossover operator and a simple mutation operator are applied for nodes and leaves [43]. Many clones will survive through a manual selection process of those textures more similar to those in the original object (in this case a croissant), and the surviving mathematical code will produce offspring. Next, using its generic formulations, the algorithm will create more individual clones and so on, until the user feels necessary.

5.2 Convergence and diversity

The search process for possible solutions is critical for achieving both convergence and diversity. Since the algorithm has been designed for its use as IEC, the convergence of the automated form has been checked, and the subsequent convergence tests would be temporarily complicated. Three graphical representations corresponding to the Matyas, SixHump Camel and Easom [44] functions within a range of [-1, 1] are used as targets. These functions are well known for optimisation and are widely used.

We chose to determine the fitness of each individual to depict the target surface in the point-cloud, using both average and the standard deviation from each point in the cloud instead of using a human subject to selection individually for the evolutionary method. The fitness function exponentially penalises surfaces where at least one point is substantially distant from the goal. In this sense, surfaces are given priority in order to minimise the average difference between the points of their representation.

For each target surface there were 50 independent runs performed to verify the stochastic stability of the convergence. Fig. 11 reflects an estimation of the wellbeing of the best individual. Results confirm the convergence with the final fitness < 0.005 is achieved. Giving the user visual alternatives is important from a variety perspective. Therefore it is important to preserve the diversity of each population, in the sense that there is no visually identical individual to another. Fig. 12 represents differences of the final population obtained for each target function between each individual. No zero-case value is reached as notice which would indicate that two individuals are visually the same.

We present a more exhaustive statistical analysis of the results obtained by the algorithm in terms of the error shown with respect to each of the objective functions following the methodology proposed by Derrac *et al.* [45] and Fernandez-Lozano *et al.* [46].

We used a Shapiro–Wilk test [47] with the null hypothesis that our results, with respect to the average surface-error, would follow a normal distribution. We obtained a p-value of 0.0000 and W=0.705013. Since p-value<.01, we reject the null hypotheses. It is assumed that the data is not normally distributed. Once this result was obtained, it was confirmed that we needed to use a non-parametric test to compare the three models.

Table 2 Post-hoc pairwise multiple comparison tests for Matyas, SixHump and Easom functions

Method	Adjuste by	Pair 1	Pair 2	p-value
Conover	FWER method	Matyas	SixHump	1.0309×10^{-57}
Conover	FWER method	SixHump	Eason	1.0309×10^{-57}
Conover	FWER method	Matyas	Eason	7.0092×10^{-86}
Conover	FDR method	Matyas	SixHump	5.1547×10^{-58}
Conover	FDR method	SixHump	Eason	5.1547×10^{-58}
Conover	FDR method	Matyas	Eason	7.0092×10^{-86}
Nemenyi	no adjustment	Matyas	SixHump	1.7149×10^{-6}
Nemenyi	no adjustment	SixHump	Eason	0.000002
Nemenyi	no adjustment	Matyas	Eason	2.3647×10^{14}

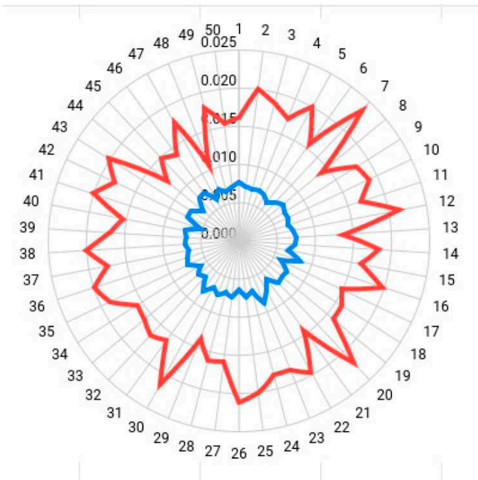

Fig. 13 *Comparison of the error between the COA algorithm and the one proposed for the Sphere function in 50 independent experiments*

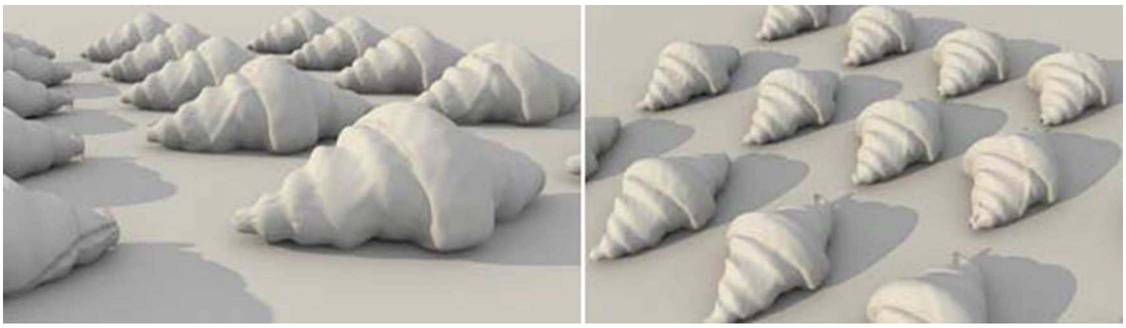

Fig. 14 *Croissant originated from the genetic morphogenesis algorithm. Diversity was achieved, since each mutant surface is slightly different*

In this case, since we have a total of 50 repeated-measures we need to use a Friedman Test [48]. The null hypothesis of the Friedman test is that there is no significant difference across functions. We obtained a p-value -1.9287×10^{-22} for rejection of the null hypothesis, that all samples are from the same distribution. The test p-value indicates whether at least one of the multiple groups (samples) is significantly different (but does not reveal which sample/group is different).

Post hoc pairwise multiple comparison tests are conducted to discern which of the pairs have significantly differences. Two of many possible post-hoc tests are conducted: the methods of (i) Conover and (ii) Nemenyi [49]. For the (i) Conover method, the p-value is adjusted in two ways, first according to the family-wide error rate (FWER) procedure of Holm, and next by the false discovery rate (FDR) procedure of Benjaminyi–Hochberg [49].

According to the p-values shown in Table 2 we can conclude that all samples are significantly different.

As discussed in the introductory section, one of the most recent alternatives is the COA [19]. This algorithm is available to the scientific community with a basic example, a sphere. As it has been done with the Matyas, SixHump and Easom functions, 50 different experiments have been performed using both the COA algorithm and the one proposed in this article.These results can be seen graphically in Fig. 13, in which the proposed algorithm improves the COA algorithm in the example proposed by its authors.

5.3 Genetic morphogenesis process

A croissant was used as an example, but this method could be used to model a wide range of existing artifacts or to produce genetically new samples of a design, all different but sharing similar internal structures (Figs. 14 and 15). Using this algorithm not only random surface deformations have been achieved, but in a manner somewhat close to what a baking process entails in real life, an infinite number of similar solutions have also been achieved.

It is worth mentioning that the algorithm functions with point clouds. Basically it involves translating NURBS geometry first into a point cloud and then implementing a mathematical genetically determined formulation on all points that are then reconstructed into NURBS geometry. A more accurate representation of the original sample resulted from the use of point geometry to produce the first sample of the morphogenesis process.

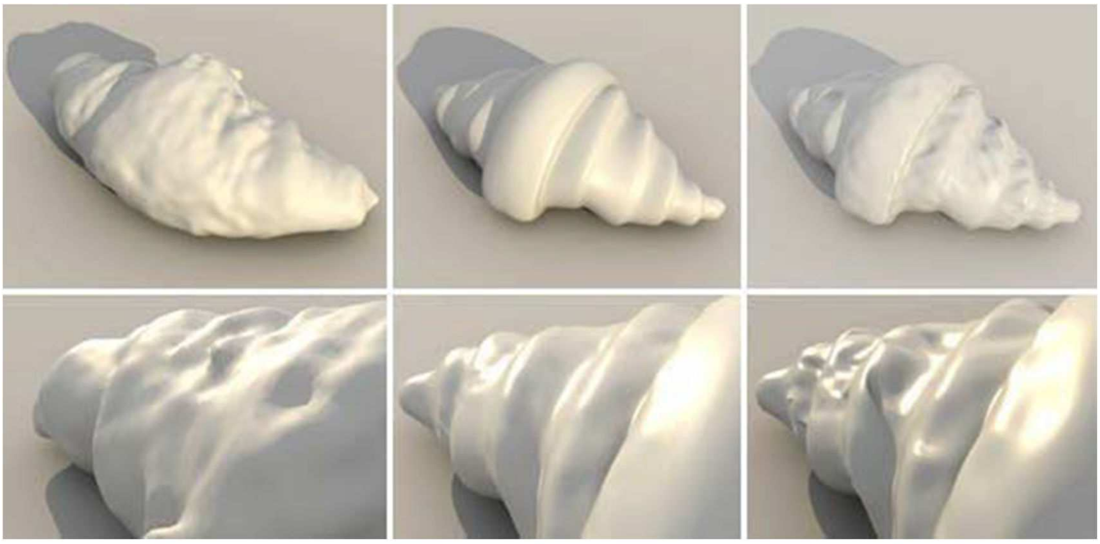

Fig. 15 *From left to right*
(a) Initial structure of cloud lines, ***(b)*** Improved geometry of NURBS, ***(c)*** The texturing point of morphogenesis

6 Conclusions

Similar rendering results were obtained from a representational point of view, using PBR and polygon mesh rendering (Figs. 8 and 9), with some variations. The point-based scanning and rendering is more similar to the geometry originally sampled, but it has limitations in terms of further enhancing or manipulating the image. On the other hand, VRay Mesh rendering allows for more manual control of light intensity and maps, obtaining a more visually appealing visualisation effect by artificially enhancing both representational and geometrically the original.

From a geometrical point of view, the resulting shapes are much more similar to the original sample, including the imperfections, but also have some inaccuracy from the scan itself, as the laser beam could not hit some points on the surface, as well as reflections (Fig. 15*a*).

For example, the manual modelling of the croissant, using NURBS and Meshes, has led to a more perfect model, but its geometry further deviates from the original one because it is too smooth and lacks imperfections (Fig. 15*b*). Compared to [27], it rationalised the original object by transforming its irregular shape into a rational one.

The research objective was first to find the best way to reliably obtain an accurate representation of an original object and then to genetically alter it through processes of morphogenesis. The invention of a genetic evolutionary morphogenesis algorithm has overcome some of the initial challenges to be able to obtain diversity by generating surface irregularity; that has also been done by an evolutionary process (Fig. 15*c*). The unique feature of our modelling process is the ability of users to alter and evolve the point cloud through a genetic recombination of surviving samples, not by changing the parameters.

The use of point clouds, as indicated by the GA, seems acceptable from the bottom up for a design process based on self-organising processes. This is because such structures are composed of simple unconnected components, with simple rules of interaction that would cause complex curvature in this case. Due to its much simpler structure, some of the drawbacks of NURBS and Polygon Meshes were overcome by using point-based geometry combined with the GA, for both accurate representation and further morphogenicity.

7 References

[1] Vasari, G., Foster, J.: '*Lives of the most eminent painters, sculptors, and architects*'. Number v. 3 in Lives of the Most Eminent Painters, Sculptors, and Architects (George Bell, Italy, 1872)

[2] Dürer, A., Strauss, W.L.: '*The painter's manual: a manual of measurement of lines, areas, and solids by means of compass and ruler assembled by albrecht Dürer 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)

[3] Hill, M.: 'Practical and symbolic geometry in Borromini's san carlo alle quattro fontane', *J. Soc. Architectural Historians*, 2013, **72**, (4), pp. 555–583

[4] Rocker: 'Interface: Between analog and digital systems', 2010

[5] Monge, G., Halbleder, L., Einband, Z.A.: '*Géométrie descriptive: leçons données aux Écoles normales, L'an 3 de la République*' (Baudouin, Imprimeur du Corps législatif et de l'Institut national, France, 1798)

[6] Dunn, N.: '*Digital fabrication in architecture*' (Laurence King Publishing, Boston, MA, USA, 2012)

[7] Marcos, C.L.: 'Complexity, digital consciousness and open form: A new design paradigm', 2010

[8] Turing, A.: 'The chemical basis of morphogenesis', *Phil. 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, NY, USA, 2012)

[10] Leach, N.: '*Digital morphogenesis*' (Wiley and Sons, New York, USA, 2009)

[11] DeLanda, M.: '*Deleuze and the use of the genetic algorithm in architecture*' (Wiley and Sons, New York, USA, 2005)

[12] Andersen, P., Salomon, D.: '*The architecture of patterns*' (W. W. Norton, UK, 2010)

[13] Dokeroglu, T., Sevinc, E., Kucukyilmaz, T., *et al.*: 'A survey on new generation metaheuristic algorithms', *Comput. Ind. Eng.*, 2019, **137**, pp. 106040

[14] Boussaï, D.I., Lepagnot, J., Siarry, P.: 'A survey on optimization metaheuristics', *Inf. Sci.*, 2013, **237**, pp. 82–117

[15] Goldberg, D.E.: 'Genetic algorithms in search', *Optimization, and Machine Learning*, 1989, **15**, (3), pp. 95–99

[16] Yang, X.S.: '*Nature-inspired metaheuristic algorithms*' (Luniver press, 2010)

[17] Mahdavi, S., Shiri, M.E., Rahnamayan, S.: 'Metaheuristics in large-scale global continues optimization: a survey', *Inf. Sci.*, 2015, **295**, pp. 407–428

[18] Shayanfar, H., Gharehchopogh, F.S.: 'Farmland fertility: a new metaheuristic algorithm for solving continuous optimization problems', *Appl. Soft Comput.*, 2018, **71**, pp. 728–746

[19] Pierezan, J., Dos, Santos: 'Coyote optimization algorithm: a new metaheuristic for global optimization problems'. 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 2018, pp. 1–8

[20] Pierezan, J., Maidl, G., Yamao, E.M., *et al.*: 'Cultural coyote optimization algorithm applied to a heavy duty gas turbine operation', *Energy Convers. Manage.*, 2019, **199**, pp. 111932

[21] Klein, C.E., Mariani, V.C., dos Santos-Coelho, L.: 'Cheetah based optimization algorithm: a novel swarm intelligence paradigm'. ESANN, Bruges, Belgium, 2018

[22] Klein, C.E., dos Santos-Coelho, L.: 'Meerkats-inspired algorithm for global optimization problems'. ESANN, Bruges, Belgium, 2018

[23] Yapici, H., Cetinkaya, N.: 'A new meta-heuristic optimizer: pathfinder algorithm', *Appl. Soft Comput.*, 2019, **78**, pp. 545–568

[24] de Vasconcelos-Segundo, E.H., Mariani, V.C., dos Santos-Coelho, L.: 'Metaheuristic inspired on owls behavior applied to heat exchangers design', *Thermal Science and Engineering Progress*, 2019, **14**, pp. 100431

[25] Moosavi, S.H.S., Bardsiri, V.K.: 'Poor and rich optimization algorithm: a new human-based and multi populations algorithm', *Eng. Appl. Artif. Intell.*, 2019, **86**, pp. 165–181

[26] Sulaiman, M.H., Mustaffa, Z., Saari, M.M., *et al.*: 'Barnacles mating optimizer: a new bio-inspired algorithm for solving engineering optimization problems', *Eng. Appl. Artif. Intell.*, 2020, **87**, pp. 103330

[27] Miralles, E.: '*How to lay out a croissant*' (El Croquis, Spain, 2000)

[28] Kuhlo, M.: '*Architectural rendering with 3ds max and V-ray: photorealistic visualization*' (Taylor & Francis, UK, 2013)

[29] Levoy, M., Whitted, T.: 'The Use of Points as a Display Primitive'. UNC report'. University of North Carolina, Department of Computer Science.' 1985

[30] Rusinkiewicz, S., Levoy, M.: ‘Qsplat: a multiresolution point rendering system for large meshes’. Proc. 27th Annual Conf. on Computer graphics and interactive techniques, New Orleans, LA, USA, 2000, pp. 343–352

[31] Botsch, M., Spernat, M., Kobbelt, L.: ‘Phong splatting’. Proc. First Eurographics Conf. on Point-Based Graphics (SPBG'04), Aire-la-Ville, Switzerland, Switzerland, 2004, pp. 25–32

[32] Zwicker, M., Räsänen, J., Botsch, M., *et al.*: ‘Perspective accurate splatting’ 2004

[33] Akenine-Möller, T., Haines, E., Hoffman, N.: ‘*Real-time rendering*’ (A K Peters, Alabama, USA, 2008, 3rd edn.)

[34] Holland, J.H.: ‘*Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*’ (University of Michigan Press, USA, 1975)

[35] Goldberg, D.E.: ‘*Genetic algorithms in search, optimization, and machine learning*’. Artificial Intelligence (Addison-Wesley Publishing Company, Massachusetts, USA, 1989)

[36] Dawkins, R.: ‘*‘The blind watchmaker’. penguin science.*’ (Penguin Books, UK, 1991)

[37] Todd, S.J.P., Latham, W.: ‘*Evolutionary art and computers*’ (Academic Press, New York, NY, USA, 1992)

[38] Bentley, P.: ‘*Evolutionary design by computers*’. Number parte 1 in Evolutionary Design by Computers (Morgan Kaufman Publishers, Massachusetts, USA, 1999)

[39] Machado, P., Cardoso, A.: ‘NEvAr — the assessment of an evolutionary art tool’. Symp. on Creative and Cultural Aspects and Applications of AI and Cognitive Science (AISB'00), Birmingham, UK, 2000

[40] Machado, P., Cardoso, A.: ‘All the truth about NEvAr’, *Applied Intelligence, Special Issue on Creative Systems*, 2002, **16**, (2), pp. 101–119

[41] Pazos-Perez, R.I.: ‘Assisted surface redesign by perturbing its point cloud representation’, *IET Softw.*, 2018, **12**, (6), pp. 251–257. Available from: http://digital-library.theiet.org/content/journals/10.1049/iet-sen.2016.0298

[42] Guilford, J.P.: ‘*‘The nature of human intelligence’. McGraw -Hill series in psychology*’ (McGraw-Hill, New York, NY, USA, 1967)

[43] Koza, J.R.: ‘*‘Genetic programming: on the programming of computers by means of natural selection’.A Bradford book*’ (Bradford, Massachusetts, USA, 1992)

[44] Molga, M., Smutnicki, C.: ‘Test functions for optimization needs’, Test functions for optimization needs, 2005

[45] Derrac, J., García, S., Molina, D., *et al.*: ‘A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms’, *Swarm and Evolutionary Computation*, 2011, **1**, (1), pp. 3–18. Available from: http://www.sciencedirect.com/science/article/pii/S2210650211000034

[46] Fernandez-Lozano, C., Gestal, M., Munteanu, C.R., *et al.*: ‘A methodology for the design of experiments in computational intelligence with multiple regression models’, *Peer J.*, 2016, **e2721**, pp. 1–22

[47] PIRO, S.H.A, WILK, S.S., M.B.: ‘An analysis of variance test for normality (complete samples)’, *Biometrika*, 1965, **52**, (3-4), pp. 591–611. Available from: https://doi.org/10.1093/biomet/52.3-4.591

[48] Friedman, M.: ‘A comparison of alternative tests of significance for the problem of *m* rankings’, *Ann Math Statist*, 1940, **11**, (1), pp. 86–92. Available from: https://doi.org/10.1214/aoms/1177731944

[49] Bhapkar, V.: ‘Practical nonparametric statistics’, *Technometrics*, 2012, **14**, pp. 977–979