UNIVERSIDADE DA CORUÑA

DEPARTAMENT OF INFORMATION AND COMMUNICATION

TECHNOLOGIES

SCHOOL OF COMPUTER SCIENCE

DOCTORAL THESIS

Blurring the Boundaries Between Real and Artificial in Architecture and Urban Design through the Use Artificial Intelligence.

Directors

Dr. Juan Ramón Rabuñal Dopico

Dr. Adrián Carballal Mato

Doctoral Candidate

Rafael Iván Pazos Pérez

A Coruña, 7th of July 2017

Mr. Juan Ramón Rabuñal Dopico,

Associate Professor of Universidade da Coruña, and

Mr. Adrián Carballal Mato,

Associate Professor of Universidade da Coruña,

Certify that:

The thesis 'Blurring the Boundaries Between Real and Artificial in Architecture and Urban Design through the Use Artificial Intelligence', was done by Mr. Rafael Iván Pazos Pérez, under our supervision in the Department of Information and Communication Technologies of the Universidade da Coruña, and constitutes the Candidate Thesis for the Doctoral Degree in Computer Science of the Universidade da Coruña.

A Coruña, 7th of July 2017

Signed: *Dr.* Juan Ramón Rabuñal Dopico Signed: *Dr.* Adrián Carballal Mato

ACKNOWLEDGEMENTS

ACKNOWLEDGEMENTS

I would like to acknowledge and thank to all the people that have supported me over that past years, both academically and professionally. I would like to specially thank and dedicate this thesis to my parents, whom have always supported and encouraged me during my academic and professional career. I would also like to particularly thank to Juan Ramón Rabuñal Dopico, Alejandro Pazos Sierra and Adrián Carballal Mato, for their motivation, support and dedication during the development of this work, at the School of Computer Science of the Universidade da Coruña.

Some of the ideas developed through this doctoral thesis date back to 2001, during my Master degree studies at the at the Graduate School of Architecture at Columbia University, in New York. During that period, I was introduced to the concepts of self-organization by professors Manuel Delanda and Kramer Woodward. During that period, I was also introduced to digital design by Joseph Kosinki (Director of the animation films 'Tron Legacy' and 'Oblivion').

Also, I would like to thank my former architecture school professors at Universidade da Coruña, from whom I learned some of the basic architecture and urban design concepts that have contributed to this thesis: Carlos Quintans, Jose Ramon Alonso Pereira, Juan Luis Dalda, Cebrian-Tello, Javier Peña, Fernando Blanco and Jesus Irisarri.

Special thanks to my professional mentors in New York, who guided me at the start of my professional career, and have somehow contributed with their guidance over the past years to the work on this thesis: Steven Apking and Peter Eisenman.

Also, I would like to thank to former colleagues, friends and partners in A Coruña, New York, Seoul and Tokyo, as they have contributed to my professional and personal development with their support: Nick cotton, Verena Haller, Jesus Colao, Max Sanjulian, Emilio Rodriguez Blanco, Pablo Otero, Omar Alvarez, Ashraf Abdallah, Eunice Kim, Inha Jung, Sookjun Roh, Karen Kim, Insun Cho, Jason Horton, Hojung An, Gorka Blas, Satoshi Yoshida, Atsumi Fujita, Joe Ginsberg and Mikako Oshima. Without their great help and support, the work in this doctoral thesis would not have been possible.

ABSTRACT / RESUMEN / RESUMO

ABSTRACT

This doctoral thesis explores the use of three-dimensional (3D) technologies for architectural representation and modelling of both 'real world' and 'artificially generated' 3D objects. Current 3D architectural representation has reached high levels of reality, to the extent that it's often hard to distinguish between 'pictures from the real world' and 'artificially generated' 3D renderings. The thesis will make use of the latest available 3D technologies in combination with artificial intelligence (AI) processes to increase the visual realism and geometrical precision of 3D models.

The first line of research relates to architectural representation and visualization, by exploring the use of 'Light Imaging Detection And Ranging' (LIDAR) technology and proposing a point-based rendering (PBR) methodology, to seamless merge models obtained directly from the 'real world' with 'artificially generated' ones.

The second line of research is related to geometrical architectural modelling, and proposes the use of evolutionary computation and self-organization logic to achieve more geometrical realism and accuracy in the 3D modelling process, by exploring the idea of auto-form generation.

The research consists of three case studies, and algorithms are proposed for each one. The first one is related to 3D visualization through LIDAR scans and PBR rendering, the second to geometrical generation through evolutionary morphogenesis and the third one to human made self-organized systems (cities). From the results obtained from each case study final conclusions will be draw. The final objective is to determine the efficiency of using point-based technologies and artificial intelligence as a methodology to further blur the boundaries between 'real world' 3D models and 'artificially generated' ones.

RESUMEN

La Tesis explora el uso de tecnologías tridimensionales (3D) para la representación arquitectónica y modelado 3D, tanto de objetos 'reales' como de objetos generados 'artificialmente'. El uso de tecnologías 3D ha permitido a la representación arquitectónica alcanzar niveles de realidad similares a fotografías, hasta tal punto que a menudo es difícil distinguir entre imágenes reales y representaciones 3D generadas artificialmente por ordenadores. La Tesis estudia el uso de las últimas tecnologías 3D disponibles, combinadas con procesos de inteligencia artificial como una forma que permita aumentar el nivel de realidad visual y exactitud geométrica de modelos 3D.

Para ello la Tesis sigue dos líneas principales de investigación. La primera está relacionada con la representación y visualización arquitectónica, explorando el uso de la **tecnología láser para detectar imágenes y medidas (LIDAR) y** proponiendo una metodología de visualización mediante el uso de técnicas de **infografías basadas en puntos (PBR)** con el fin de combinar directamente modelos 3D obtenidos directamente del 'mundo real' y modelos 3D 'generados artificialmente'. La segunda línea de investigación está relacionada con el modelado de geometrías arquitectónicas. Mediante el uso de computación evolutiva y procesos de auto-organización con el fin de lograr un mayor grado de realismo y exactitud en el modelado 3D, mediante la introducción de parámetros del mundo real en los algoritmos de morfogénesis evolutiva, explorando así la idea de la auto-generación de formas. La Tesis propone a través de tres casos experimentales, algoritmos para cada uno de ellos. El primero está relacionado con la visualización, el segundo con la morfogénesis geométrica y el tercero con sistemas complejos auto-organizados realizados por el hombre (ciudades). A partir de la evaluación de los resultados obtenidos de cada caso experimental, la tesis extraerá conclusiones finales sobre el uso de técnicas de inteligencia artificial como metodología para difuminar los límites entre los 'modelos de 3D del 'mundo real' y los 'generados artificialmente'.

RESUMO

A Tese explora o uso de tecnoloxías 3D para representación arquitectonica e modelaxe 3D, tanto de obxectos 'reais', coma de obxectos xerados artificialmente. O uso de tecnoloxías 3D na representación arquitectónica pode acadar uns niveis de realidade similares a fotografías, na medida en que moitas veces é difícil distinguir entre imaxes reais e representacións 3D xeradas artificialmente por ordenadores. A tese estuda o uso das últimas tecnoloxías 3D dispoñibles, combinadas con procesos de intelixencia artificial como un xeito de aumentar o nivel de realidade visaual e exactitude xeometrica de modelos 3D.

Esta tese segue dúas liñas principais de investigación. A primeira está relacionada coa representación e visualización arquitectónica, explorando o uso das tecnoloxías láser para detectar imáxes e medidas (LIDAR) e propoñe unha metodoloxía de visualización mediante o uso de infografías basadas en puntos (PBR) para combinar directamente modelos 3D obtidos directamente do 'mundo real' e modelos 3D 'xerados artificialmente'.

A segunda liña de investigación está relacionada coa modelaxe de xeometrías arquitectonicas. Usando procesos de computacion evolutiva e auto-organización a fin de acadar un maior grao de realismo en modelaxe 3D, a través da introdución de parámetros do mundo real en algoritmos evolutivos de morfoxénese, deste xeito explorando a idea de auto-xeración de formas.

A tese propón a través de tres casos experimentais, algoritmos para cada un deles. O primeiro esta relaccionado coa a visualizacion, o segundo coa morfoxénese xeométrica eo terceiro cos sistemas complexos auto-organizados feitos polo home (cidades). A partir da avaliación dos resultados de cada caso experimental, a tese extraerá conclusións finais sobre o uso da intelixencia artificial como unha metodoloxía para difuminar as fronteiras entre os modelos 3D reais e os xerados artificialmente.

TABLE OF CONTENTS

LIST OF ABREVIATIONS

3D: Three-Dimensional

AD: Anno Domenico (Year)

ADF: Automatically Defined Functions

AI: Artificial Intelligence

ANNs: Artificial Neural Networks

API: Application Program Interface

BC: Before Christ (Year)

BIM: Building Information Modeling

Bldgs.: Buildings

CA: Cellular Automata

CAD: Computer-Aided Design

CATIA: Computer Aided Three-Dimensional Interactive Application

CUCs: Corporate Urban Centers

CG: Computer Graphics.

D&C: Divide and Conquer

DEM: Digital Elevation Model

DNA: Deoxyribonucleic acid

EC: Evolutionary Computation

F-distribution: Fisher-Snedecor distribution

F-Test: Fisher-Snedecor Test

GA: Genetic Algorithm

GDP: Gross Domestic Product

GIS: Geographic Information System

GL: Graphics Library

GP: Genetic Programing

GPU: Graphics Processing Unit

IEC: Interactive Evolutionary Computation

IGA: Interactive Genetic Algorithm

KD-tree: K dimensional tree

Km: Kilometer

LIDAR: Light Imaging Detection and Ranging

LISP: List Processing

LOD: Level of Detail

NEvAr: Neuro Evolutionary Art

m: Meter

m2: Square meter.

OpenGL: Open Graphics Library

OS: Operating System

PBR: Point-based rendering

PC: Personal Computer

PPU: Points per unit

MEL: Maya Embedded Language

MSAA: Multisampling anti-aliasing

NURBS: Non-Uniform Rational Basis-Spline

R-squared: Coefficient of Determination

RT: Real Time

S: Slenderness

SUS: Stochastic Universal Sampling

UV: (x,y) coordinates of a surface texture mapping

UVW: (x,y,z) coordinates of a solid texture mapping

V-V mesh: Vertex to Vertex mesh

WWII: World War Two

TST: Topographic Total Station Theodolite

INDEX OF FIGURES, TABLES AND EQUATIONS

INDEX OF FIGURES

INDEX OF TABLES

INDEX OF EQUATIONS

1 INTRODUCTION AND OBJECTIVES

1.1 Introduction and Contextualization.

Technological advances in the field of three-dimensional (3D) technologies have recently leaded to a whole revision of the postulates of architectural representation. Recent 3D techniques allow for accurate representation of real environments and the generation of virtual spaces, by blurring the limits between 'real' and 'non-real' spaces. Some of the latest developments include 3D 'Light Imaging Detection and Ranging' (LIDAR) scanners that allow 3D scanning of real environments to be later processed in a computer. Other recent developments are 3D printing devices that allow the printing of three dimensional objects from a 3D virtual model.

LIDAR scanner technologies use a high precision laser beam (within 1 mm precision range), able to analyze a real object or environment taking data from its geometry, color and texture. This data can be used to reconstruct the scans into 3D models, with a large variety of uses and applications. This technology allows for the compilation of accurate data within a very reasonable timeframe. The format used by the scanners is typically referred as a Point cloud, consisting on a database of vertex referred to a coordinate system. The vertex is defined by its coordinates (X, Y, Z) that represent the external surfaces of the scanned objects. The points typically also contain other information, such as color, reflectivity index and illumination levels.

This technology allows for a diverse range of applications in architecture, such as the reproduction of architectural details, topographic definition of buildings or civil works, 3D cataloguing of historical heritage and representation of public spaces. Other applications are the use of the data for its manipulation into new designs, historical renovations or morphogenesis processes.

The advantages of 3D LIDAR scanning are currently limited by the size of the scans, which typical are difficult or impossible to be processed by current personal computers, jeopardizing its applications by common users and in the professional world. Currently new point-based computation techniques are under development; these new procedures already allow for the visualization of millions of points captured by a LIDAR scan. The further development of point-based techniques opens new possibilities for the use of LIDAR computation capabilities. Furthermore, points are an ideal format for the application of artificial intelligence (AI) based algorithms, such as self-organization, morphogenesis and evolutionary processes. The use of point geometry in combination with AI opens the possibility of new architectonical methodologies that could lead to major changes in the computational methodological postulates of architectural design and representation, by further blurring the boundaries between the ‘real’ and 3D ‘artificially generated’ world.

The use of AI and evolutionary computation (EC) allow for the generation of adaptive self-organizing systems and geometrical morphogenesis processes, bringing ‘artificial generated’ models closer to the ‘real world’ geometries. Turing (1952) noted on ‘The Chemical Basis of Morphogenesis’ regarding the recurring patterns of growth in flowers, how by using mathematical tools complex organisms could assemble itself without any master planner calling the shots. It is important to note the fact that when Turing developed his mathematical formulas to explain the recurring patterns of morphogenesis in plants he idealized cells as points. Topological geometry was more appropriate for the mathematical formulation of cell growth in plants. In Turing’s words: ‘Most of organisms develop from a pattern into another, rather than from homogeneity into a pattern, mostly as the result of a bottom-up self-organizing process, governed by simple recurring patterns and local interactions, basically a morphogenesis process’.

Turing example of growth leading to a specific structure resulting into recurring but always different geometries is actually like some man-made processes, and more in particular urban growth. As Al Sayed et Turner (2012) have noted how most of computers models that simulate urban growth, are developed based on mechanisms that characterizes biological systems, rather than spatial systems.

Holland (1975 &1998) explored the way simple rules could lead to complex behavior, he took the logic of Darwinian evolution and built it on what he called a genetic algorithm. Computer codes of ones and zeros and coiled strains of DNA constitute the genotype, or genetic code. The higher form of behavior that those codes produce is what is called the phenotype, such as growing or multiplying, for instance. The evolutionary processes consist in the process of randomly combine pools of genes, in this case codes, and depending on the rate of succeed of the new solutions and behaviors, the successful variations will pass to the next generation (Johnson 2001). Random mutations also play a key role in the evolutionary process, by providing variants into the system. Genetic algorithms (GA) offer new possibilities for the computer self-generation of architectural and urban geometries.

1.2 Objectives.

The objective of this thesis is to experimentally evaluate the use of point-based rendering techniques and evolutionary algorithms for 3D visualization and for AI processes, more in particular morphogenesis, to achieve more integration in between 3D models from the 'real world' and 'artificially generated' 3D models.

Photogrammetric and LIDAR scanners can accurately scan urban environments both as polygon meshes or point clouds, depending on the technology used. This process constitutes a considerable scientific advance, as it will suppose a new research field with practical applications for architecture and urban design, both by the use of point-based representation and point-based AI processes.

The research will explore the use of evolutionary algorithms. By imputing recurring morphological patterns to the algorithms, and applying them to a 3D model of an architectural object or to a city's geometry to automatically generate morphogenesis processes and self-organized growth. The parameters will be generated from based pre-existing data to later auto generate thousands of possible geometries, based on the patterns previously identified.

According to Weinstock et al (2004) 'Emergence and morphogenesis are terms that relate to natural processes of form generation'. Furthermore, in morphogenesis, form and behavior emerge from evolutionary processes that interact with and exchange information with the natural system and its environment (Oxman 2010).

Ultimately the goal of the thesis is to study the use of algorithms for both visualization and morphogenesis of 3D geometries, to achieve greater degree of realism in 3D processes, by merging 'real' and 'artificially' generated objects.

1.3 Thesis Structure.

This doctoral thesis is organized in chapters, starting with a brief description of the context and objectives. Then the thesis will explain and describe the fundamental concepts and ideas that will be used to test the thesis hypothesis. The methodology will be explained to then develop the main research work through several case studies. Each one of the case studies will test each sub-hypothesis. Finally, conclusions will be drawn and further lines of research proposed.

Chapter 1. Introduces the issues to be researched, the objectives, historical background, and the aim of the thesis.

Chapter 2. Explores the current concepts and fundamentals that will be used in the research. The basic fundamentals of 3D modelling, 3D formats, LIDAR scanning, point-based rendering (PBR), AI, genetic algorithms, morphogenesis, and self-organization.

Chapter 3. Explains the main hypothesis of the thesis, the objectives and goals to be achieved.

Chapter 4. Explains the research methodology used, as well as the strategies that will be followed to test the sub-hypothesis.

Chapter 5. Explores through experimental case studies the combination of LIDAR scan data with manually generated 3D models, using PBR and will propose two visualization algorithms.

Chapter 6. Explores the generation of natural irregularity and diversity in 3D models by proposing an evolutionary algorithm.

Chapter 7. Proposes an evolutionary algorithm to predict the behavior of complex man made self-organizing systems such as the vertical growth in cities.

Chapter 8. Will draw final conclusions of the research, linking the individual results and evaluations from each case study.

Chapter 9. Proposes additional lines of research topics that could be further studied beyond the scope of this thesis.

Chapter 10. List the bibliography and works used as reference for the development of this thesis.

2 FUNDAMENTALS

2.1 Digital Representation Methods.

2.1.1 Fundamentals.

This sub-chapter describes the different types of 3D technologies commonly used for architectural and urban representation, as well as the most typical formats and computational entities used for 3D.

There are a wide range of software packages available in the market for 2D and 3D representation, some packages however are more widely used in the architecture and urban design industry, due mostly to commercial reasons, and not just necessarily due to its performance and capabilities. Each one of those software packages has its own extensions and file formats. Computer-Aided Design (CAD) is the first nomenclature used for the drawing making techniques involving the use of personal computers. CAD programs have its origins in Sutherland's (1963, 1965) early computer program called 'Sketchpad'. As Davis (2014) has noted, Timothy Johnson video at the Lincoln Laboratory (1964) demonstration of Ivan Sutherland's (1963) Sketchpad was the first time that someone was seen interacting and drawing with a computer. As Davis (2014) has noted: 'Johnson's actions initially resemble a designer at a drawing board, the ink replaced with light, the impression of the computer's similarity to the drawing board only broken when Johnson creates a misshapen trapezium and defiantly states: I shouldn't be required to draw the exact shape to begin with'. Since this first demonstration computer software's have evolved into modern day CAD programs. Current CAD software not only allow for digital representation but also allow for digital fabrication. Another application is the production of realistic 3D images through a process known as Computer Graphics (CG), also often referred as computer renderings.

It is important to make a distinction between 3D modeling and 3D rendering, which relate to Digital Design and Digital Representation respectively. Modeling describes the process of making a 3D shape, geometry, volume or object virtually in a computer. Currently there are four main types of 3D models: Point-based, Polygon based, Voxel based and NURBS based. Voxel based models are mainly used in bio-medical applications, and this thesis will focus exclusively on the other three formats only. The least commonly used format among the three are the point-based formats, however will be extensively used.

2.1.2 Computer Graphics

Computer Graphics, is the discipline that creates, manages and visualizes graphics using computers. Traditionally CG has been defined as the computer science discipline that is dedicated to synthesize images algorithmically with computers. Recent advances in CG have impacted a large array of industries, such as media, movies, advertising, video games, civil engineering, industrial design, medicine, manufacturing, interior design, architecture and urban design just to name a few.

The differentiated process within CG, are 3D modeling and rendering. 3D Modeling consists on the generation of 3D geometrical data on a computer, typically by polygon models, NURBS, vectors or points. Rendering is the process of generating a 2D image from 3D data (modelling), by adding lights and cameras, to be later shown on a display or printed on a surface. The images produced in the rendering process, can be used to produce still images, or to produce animations, by producing still frames on a time-line, generating 3D animations. Renderings can be divided into non-realistic renderings and photorealistic renderings. Non-realistic renderings create images and animations in 3D than differ from the real world, as an illustration, painting or drawing does. A photorealistic rendering, tries to imitate the reality as realistically and accurate as possible.

When processing a rendering, there are two types of processing. The first one is off-line rendering, which are high quality renderings that will take a computer from few minutes to days to render each frame. This type of processing is common in movies and architectural renderings for instance.

The other type is the on-line rendering, commonly known as 'Real Time Rendering' (RT). This type of processing renders fast in a way that the user can interact in real time with the images rendered, giving a sensation of continuity, typically in the range of more than 20-30 rendered frames per second. This type of rendering has lower resolution than the previous one, and it is commonly used in video games and other applications

There are wide range rendering techniques and rendering engines, which differ a lot the way they generate the image. There are some common steps that all rendering engines follow. The first step is to set up a camera position within the coordinate system. The 3D geometry will be then projected to the camera point (location), intersecting each projection ray with a plane, each intersection resulting into a pixel, which color is calculated through a wide range of rendering techniques.

The Graphics Processing Unit (GPU) pipeline is the process implemented in hardware to create images from the scene information in a highly efficient way. It is comprised of several stages were several lineal algebra operations that account for the position, rotation and scaling of the point of view, objects and lights; to finally determine the color of the pixel drawn to the screen. The frame-buffer is the final image that will be shown onscreen, after all the calculations have been made.

Regarding 3D modelling as Dunn (2012) has noted, there are two main ways of making three-dimensional forms digitally, NURBS and meshes. This thesis however will add a new one, not so commonly used in architecture, point clouds.

2.1.3 3D Modelling and Rendering.

3D modeling and rendering software's allow to for the creation and modification of three-dimensional shapes and geometries. They are widely used for architecture and urban design. There are many different software packages currently used and commercialized under different brands, some of them are open source; some of them are for commercialization purposes. The following 3D modeling software packages have been identified:

3D Slash, 3D-Coat, 3DVIA Shape, AC3D, Anim8or, Animation Master, Arc+, ArchiCAD, Art of Illusion, Autodesk AutoCAD, Autodesk 123D, Autodesk 3ds Max, Autodesk Inventor, Autodesk Maya, Autodesk Mudbox, Autodesk Revit, Autodesk, Softimage, AutoQ3D, AutoQ3D Community, Blender, BricsCAD, BRL-CAD, Bryce, Carrara, CATIA, Cheetah3D, Cinema 4D, CityEngine, Clara.io, DAZ Studio, DesignSpark Mechanical, Electric Image Animation System, EQUINOX-3D, Exa Corporation, Flux, Form-Z, fragMOTION, FreeCAD, Geomagic Freeform, Geomagic Sculpt, Geomodeller3D, Hexagon, Houdini, IronCAD, ixCube, Leapfrog3D, LightWave 3D, Massive, Metasequoia, MikuMikuDance, MilkShape 3D, Modo, Moi3D, Open CASCADE, OpenSCAD, Poser, Pro/ENGINEER, Quake Army Knife, RaySupreme, Realsoft 3D, Remo 3D, RFEM, Rhinoceros 3D, Sculptris, Seamless3d, Shade 3D, Silo, Sketchup, Socet Set, Solid Edge, SolidThinking, SolidWorks, SpaceClaim, Strata 3D, Sweet Home 3D, Swift 3D, Tekla, Structures, TinkerCad.com, TrueSpace, Unigraphics, Vectorworks, Windows 3D, Builder, Wings 3D, ZBrush, Zmodeler, ZWCAD.

This chapter will briefly describe some of the most commonly and widely used 3D modeling software packages in the architecture and urban design industry, which will be relevant to the contents of this thesis.

Autodesk AutoCAD, it is a CAD software package. It is one of the first widely used software's in the architectural industry. It was originally a 2D software, but currently also has 3D and rendering capabilities. Its strength is the production of 2D drawings, that replicate the traditional pre-computer representation methodologies. It works through vector computation, but is also able to read other formats. His main formats are DWG and DXF.

Autodesk Revit, it is a widely-used Building Information Modeling (BIM) software, which also has rendering capabilities. It is mostly oriented to architecture modelling and construction through parametric drawings. It has the advantage that several people can work simultaneously in the same drawing, and that several disciplines can be merged into the same file. Combines 2D representation and 3D modeling, being able to produce linked 2D and 3D drawings. Its file format extension is known as RVT. Autodesk Revit LT, is a lighter version of Revit oriented to architectural modeling. It doesn't allow for simultaneous team work or allow for complex curved geometry. Its extension is same as Revit, RVT.

Autodesk 3DS Max, is a polygon mesh based software oriented for 3D Computer Graphics, originally created for the animation industry, but currently widely used in architecture, interior design, urban design and video games. It can achieve hyper photorealistic renderings and animations. It allows for 3D script creation through 'Max script', also has many plug-ins available. 3DS Max is one of the most widely used software due to his great degree of quality for photorealistic 3D renderings. 3DS Max will be extensively used, in combination with the VRAY rendering plug-in.

SketchUp, it is a free open source polygon mesh software for 3D production. It has a very easy to use and intuitive interface; it is widely used for architecture, Interior

Design, engineering, Industrial Design, Videogames and movies. It allows for scripts creation with API Ruby.

CATIA, it is a widely-used software in the air-space engineering industry, its name stands for: Computer Aided Three-Dimensional Interactive Application. It was developed by the French company Dassault Systèmes. Per Bernard (2003) the software started its development in 1977 for the aeronautical industry. CATIA combines aerodynamic surfacing model with other engineering systems, such as mechanical, electrical and fluid systems. It was used in the development of the Boeing 777. Per Day (2003), the software became known to architecture when Los Angeles based architect Frank Gehry started using it for the construction and engineering of his buildings.

Rhinoceros, it is a NURBS-based 3D software, developed by Robert McNeel & Associates, that works well with AutoCAD, widely used in architecture, industrial design and multimedia industries. Due to its NURBS mathematical model, which focuses on producing mathematically precise representation of curves and freeform surfaces is a very good software to handle complex 3D irregular curvature, and non-regulated geometry. It is not a parametric software, but through the plug-in Grasshopper parametric scripts can be introduced into the software, allowing for the introduction of generative algorithms. Also through rhino script all kind of algorithms can be designed and introduced into the software, including AI and evolutionary algorithms. Rhino Script will be the main scripting tool used on this thesis. Rhinoceros 5.0. will be extensively used as it has great 3D capabilities for the generation and manipulation of 3D complex curved geometry. Also because it allows for scripting and the creation of custom algorithms, and more in particular evolutionary algorithms.

Autodesk Maya, it is a modelling and rendering software, that works very well with NURBS, and with great animation capabilities. For that reason, is widely used in for animations, video games and renderings. It was originally developed by 'Alias Systems Corporation', then acquired by 'Silicon Graphics Inc.' and then by 'Autodesk'. It has been widely use in the animation industry, including many Hollywood feature films. Maya includes many pre-animated effects to make animated water, fluids, wind, particles, grass, etc. According Sharpe et Al (2008), after the acquisition by Silicon graphics Inc., Sophia, the scripting language in Wavefront's Dynamation, was chosen as the basis of Maya embedded language (MEL), it allows for users to code their own routines, modifications and plugs-ins.

Rendering Software Packages and Plug-ins.

In addition to the default rendering engines included on each software package, there are rendering packages and plug-ins just for rendering purposes. Some of them can be installed in some of the previous modeling software packages to produce final rendering images, some of them import models previously produced in other software packages. Each software uses its own techniques for the calculation of textures, maps, lights, cameras, etc., yielding very different results and qualities. Below there is a list of the main rendering packages available in the market, some of them for commercialization, and some of open source. Below there is a list of the most relevant rending packages currently in use are:

3Delight, AIR, Appleseed, Arion, Artlantis Studio, Aqsis, Arnold, Bakery Relight, Brazil R/S, Bunkspeed SHOT, Caustic Visualizer (Brazil variant), CentiLeo, Clarisse IFX, Colimo, Corona Renderer, Cycles, FELIX Render, FinalRender SE, Flamingo, FluidRay RT, FPrime, Fujiyama, FurryBall, Gelato, Guerilla Render, IC3D Suite, iClone, Indigo Renderer, iRay, KeyShot, Kerkythea, Kray, LightWorks, LumenRT,

Lumion, LuxRender, Maxwell Render, mental ray, Mitsuba Render, moskitoRender 1, NOX, nXtRender, Octane Render, Penguin, PhotoRealistic RenderMan, pbrt, Pixie, POV-Ray, Radiance, Raylectron, Redshift, RenderDotC, Spectral Studio, Sunflow, SPEOS, TheaRender, ToView, Twinmotion, Turtle, V-Ray and YafRay.
As it is not the purpose of this thesis to explain all the different rendering techniques and software packages, only the two rendering engines that will be used for the case studies will be described:

VRAY, it is a well-known rendering plug-in developed by 'The Chaos Group', Kuhlo and Eggert, (2010). It is available to be installed in 3DSMax, Rhinoceros, Revit, Maya, Sketch Up, Softimage, Blender, Standalone, MODO, NUKE and Phoenix FD, which also happen to be some of the most commonly used modeling software in the market. VRAY can generate hyper photorealistic 3D images, of high quality.

ToView, is a Point-based rendering software developed by the research team of the Department of Information Technology at A Coruña University in 2014. The rendering engine works very efficiently for the visualization of unstructured point clouds. ToView can efficiently handle point clouds of millions of points, due to the use of splats and level of detail, LOD. This software will be use to visualize point clouds in chapter 5, through a practical research case study.

2.1.4 Generation of 3D Data from the Real World.

There are several different techniques that allow for the accurate measurement and generation of 3D data from the 'real world', these techniques are of relevant importance for the further development of this doctoral thesis. The most common methods used are: topographic stations, photogrammetry and LIDAR scanning. This sub-chapter will briefly describe those techniques, with a special emphasis on

LIDAR scanning, as it will be used later in the research. As previously stated on chapter 1, the main objective of this thesis is to propose a methodology to achieve more integration in between 3D reality and artificially generated 3D models. For such a purpose the generation of accurate 3D models from existing real world objects is of great importance for this research.

2.1.4.1 Topographic Stations (TST).

Traditional topography techniques aim to obtain point coordinates in the space to later be accurately represented. This process has been traditionally made, and still is, by using Topographic Total Station Theodolites (TST). This topographical instrument measures by an electronic theodolite angles (slopes) and distances to a specific point (Fig. 01). Using triangulations and other mathematical computations it is possible to obtain with a great degree of precision the measurement of large areas. The process is however long and complex, and requires many calculations. TST, will be not used for research, therefore won't be described in detail. Instead more recent computer based technologies will be used.

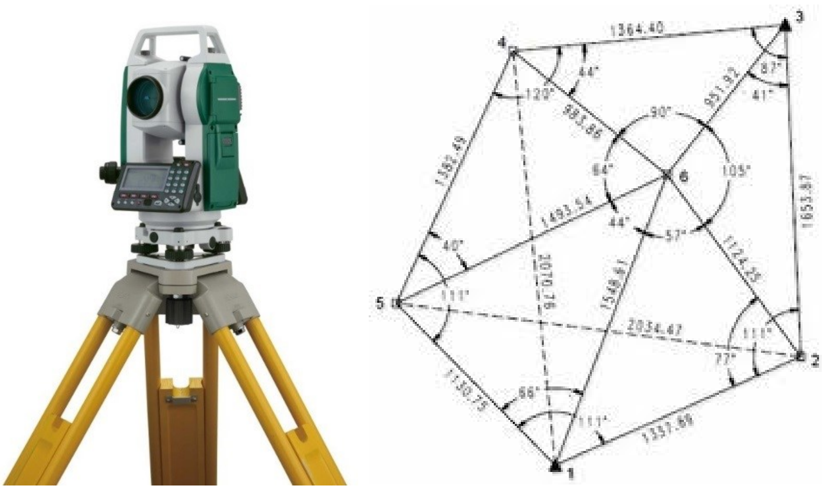


Fig. 01. On the left, Total Station 'Sokkia Set 50x'. On the right and example of triangular calculation to set the points in the space. (Source: sokkia.com).

2.1.4.2 Photogrammetry.

Per the American Society for Photogrammetry and Remote Sensing, ASPRS (2015) is: 'The art, science, and technology of obtaining reliable information about physical objects and the environment through processes of recording, measuring and interpreting photographic images and patterns of recorded radiant electromagnetic energy and other phenomena'. Photogrammetry consists in taking measurements from photographs, to typically generate a map, drawing, 3D or just taking any measurements from the real world. There are several types of photogrammetry, the standard method is to split the field based on camera location during photography. Photogrammetry is divided into close-range photogrammetry and aerial photogrammetry.

Close-range Photogrammetry is non-topographic, used for drawings, 3D models, measurements and point clouds. Everyday cameras are used to model and measure buildings, engineering structures, forensic and accident scenes, mines, earth-works, stock-piles, archaeological artifacts, film sets, etc. This type of photogrammetry is called Image-Based Modeling.

Aerial photogrammetry consists in installing a camera into a plane or drone, usually pointing vertically towards the ground. Multiple overlapping photos of the ground are taken as the aircraft flies along a flight path. The pictures will be later processed in a stereo-plotter, this type of photogrammetry is used for topographic terrain models, topographic maps and cities. The pictures can be also used to obtain the height of the objects by an automated process called Digital Elevation Model (DEM) creation, by triangulation.

The fundamental principle used by photogrammetry is triangulation. By taking photographs from at least two different locations, 'lines of sight' or 'rays' can be developed from each camera to points on the object, in a similar way that a TST does. By simple triangulation 3D dimensional coordinates can be precisely determined, by mathematically intersecting converging lines. To do that a process called 'resection' is necessary, which consists in determine the exact position and orientation of the camera when each picture was taken, the more points that are used for the resection process the more accurate the information obtained will be. The process of producing 3D coordinates from the photographs is referred as 'the Bundle Adjustment', which consist in obtaining the final 3D coordinates of a point, by triangulating the target points and the camera positioning (Fig. 02).

For such a purpose computer algorithms are used to make the calculations and to minimize errors, the bundle adjustment process is often done by using the Levenberg–Marquardt algorithm. According to Lourakis (2005), The Levenberg-Marquardt (LM) algorithm is an iterative technique that locates the minimum of a multivariate function that is expressed as the sum of squares of non-linear real-valued functions, Levenberg (1944), Marquardt (1963). It has become a standard technique for non-linear least-squares problems, widely adopted in a broad spectrum of disciplines.

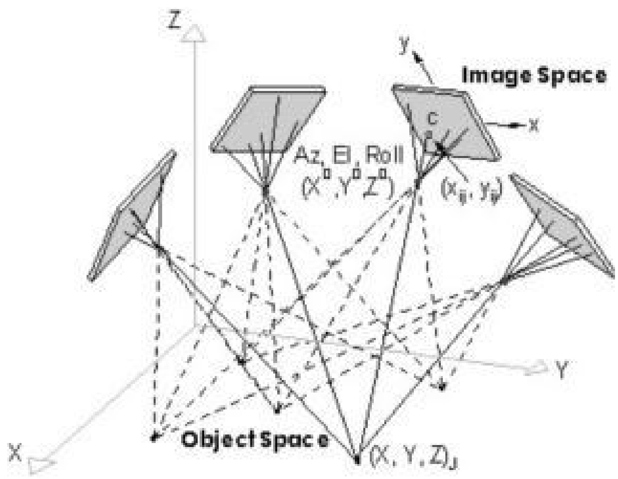


Fig. 02. Triangulation, resection and Bundle Adjustment process *(Source: www.geodetic.com).*

2.1.4.3 LIDAR: Laser Imaging Detection and Ranging.

LIDAR is a surveying method that measures distance to a target by illuminating that target with a pulsed laser light, and measuring the reflected pulses with a sensor. It is a recent technology which allow to obtain high-precision geo-referenced 3D scans from the real world in a very accurate, easy and fast way. These systems use ultraviolet, visible or near infrared light to image objects. They have precisions up to 1 mm, and applications for a wide range of technologies. The scans generate large sets of unstructured point clouds that contain large sets of referenced points, typically including color, normals and luminosity. Managing such large data sets presents some issues, due to the large amount of information, making files difficult to be managed on regular personal computers. Also, the translation of the datasets into polygons is possible, however it also poses some challenges and typically incurs in lots of inaccuracies. An example of a LIDAR device and scan is shown on Fig. 03.

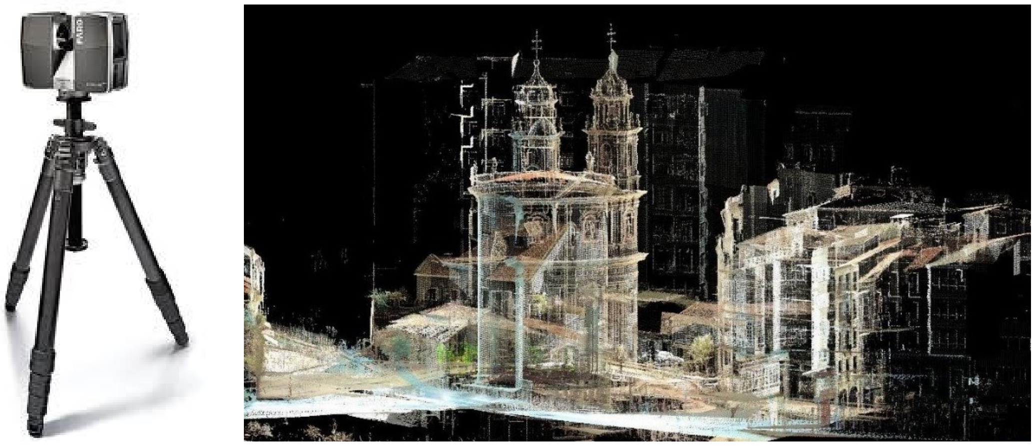


Fig. 03. On the left LIDAR device 'Faro Focus 3D'. On the right view of a LIDAR point cloud of a historical building.

The advantages of using LIDAR technology instead of the previously described technologies are numerous:

-High productivity due to short scanning time.

-High precision, and easy to visualize.

-No need to have physical contact with the objects to be measured.

-Combability with CAD, GIS and BIM software's.

-Direct generation of realistic perspectives and orthographic projections.

-Possibility to generate 3D models for 3D printing.

Through a real case study it will be briefly explained the process how the LIDAR scanner Works. The following example show the process of obtaining a LIDAR 3D scan of an interior space, Fig. 04.

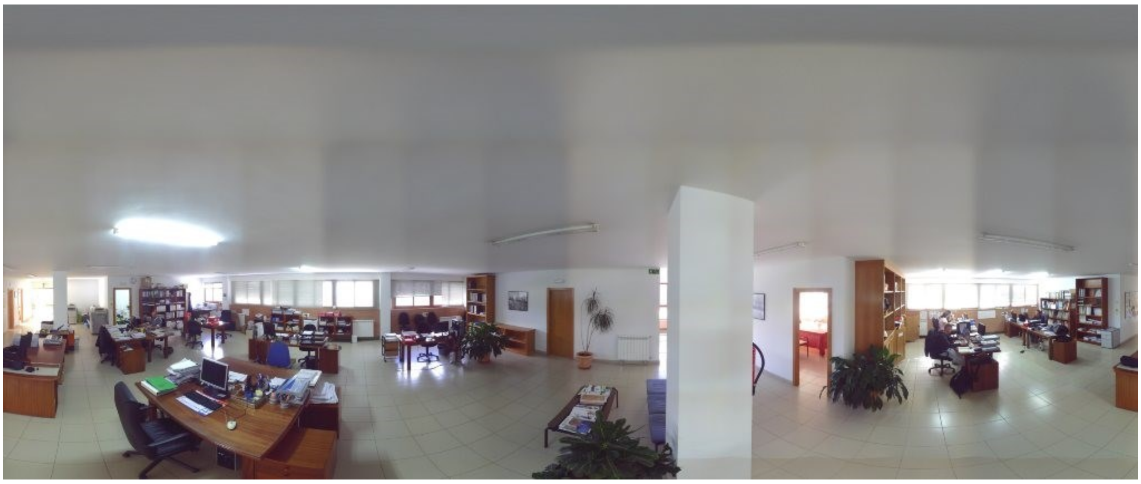


Fig. 04. 3D point-based LIDAR scan made with 'Faro Focus 3D'.

The LIDAR machine, will be placed in different locations, the scans will always be done with a certain overlap to be able to merge them later. Two methods of registration will be used. On the first one the scans will use spheres or targets which are an artificial register, (Fig. 05). The second one will be to manually record the information by using floor plans, ceilings, floors etc., or common lines and points.

Fig. 05. Software view with spheres, common points and planes.

As the plan area is relatively small, it was not necessary to use the classical topography methodologies to link the scans, but just by relying in direct measurements with laser distance-meters and flex-meters to complete the dispatch zones and to fill in the hidden areas that have not been obtained with laser scanner technology, (Fig. 06).

The postprocessing of the scans will be done with either the artificial (targets or spheres) or natural (planes, lines or common points between two or more scans). Filtering of non-necessary points will be made when need it, for this, a series of manual and semi-automatic algorithms will be executed to eliminate non-useful data.

To obtain manageable 3D models or 2D data, from the point cloud, the planes of the walls of each of the compartments will be modeled and the intersections of each of the planes will be made, thus obtaining the edges of each room. This process will generate data to be exported to CAD files and will be placed together with the data obtained from indirect measurements, to produce the final floor plans.

Fig. 06. View of the unfiltered point cloud data.

2.1.5 Point-based Rendering

The previous sub-chapter (2.1.4) explored different techniques to obtain accurate 3D data from the real world, as it is of special relevance to the research objectives. It is frequent in architecture and interior design to have to accurately draw existing buildings, structures or spaces for varied reasons, such as for renovations, extensions, historical preservation, historical heritage, or simply for the need to draw adjacent structures next to future interventions or architectural projects. LIDAR scanning is currently the fastest and more accurate methodology to accomplish such a task.

As previously mentioned, LIDAR technology uses massive point clouds as output data, and the information obtained from the scans are unstructured point clouds, extensive databases of unconnected points. The resulting 3D data can reach millions of points, making quite heavy files due to the large amount of information. As Rusinkiewicz and Levoy (2000) have noted 'Advances in 3D scanning technologies have enabled the practical creation of meshes with hundreds of millions of polygons. Traditional algorithms for display, simplification, and progressive transmission of meshes are impractical for data sets of this size'. The most commonly used architectural software packages and standard computers and graphic cards cannot manage those scans, so the information obtained is currently not practically useful for architectural offices, unless it gets preprocessed either manually or through a software package, and then converted into manageable polygon meshes, and eventually into lines.

Due to the difficulty of dealing with the large information produced by the laser scans, another approach would be using exclusively points as a methodology for architectural representation. The concept of Point-based rendering (PBR), which was originally proposed by Levoy and Whitted (1985) suggest and approach based in working directly with point clouds, instead of polygon meshes or NURBS.

As Rusinkiewicz and Levoy (2000) have noted referring to PBR: 'Computer graphics systems traditionally have used triangles as rendering primitives. In an attempt to decrease the setup and rasterization costs of triangles for scenes containing a large amount of geometry, a number of simpler primitives have been proposed'. As point clouds are massive sets of unconnected points, its translation into surfaces or lines useful for architectural representation is not straight forward, and the conversion process often incurs in lots of inaccuracies and inefficiencies. Many of the conversion techniques have focused on the placement of edges and vertices expending a lot of effort per vertex. Rusinkiewicz and Levoy (2000) first introduced the concept of QSplat which is a hierarchical multi-resolution rendering system based on points and bounding spheres, and to structure the data by using rendering algorithms, specifically defining the visibility culling, level of detail (LOD) and rendering processing. The points would be then rendered as splats (oriented surfaces), using squares, circles or Gaussians (Fig. 07).

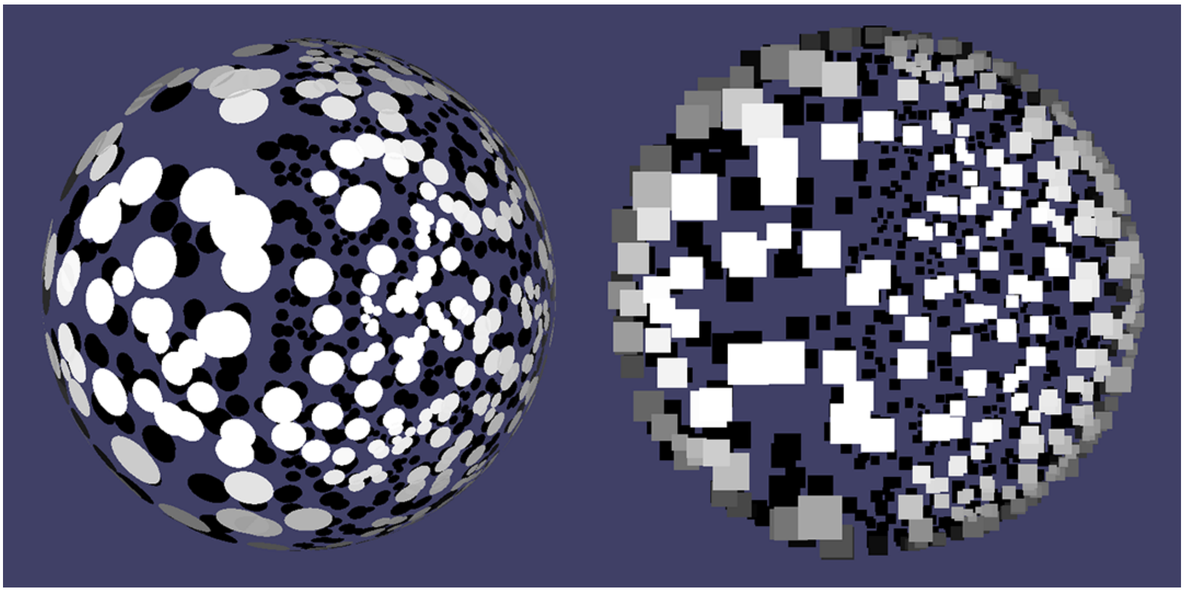


Fig. 07. On the left circular 'splats' or oriented discs perpendicular to a surface. On the right side non-oriented square 'splats'.

PBR allows for the manipulation of millions of points, as the manageability and the efficiency of points in processing and rendering is much higher than polygon-based systems, this is particular true for large models or scans of complex curved geometries.

The concept of Surface Splatting or Phong Splatting was further developed by Botsch et al. (2004) introducing the idea of blending ellipses to generate a surface, as a more efficient rendering approach for point-based geometries for detailed or complex models. By using this technique point-based surface representation becomes a more efficient alternative for rendering, as no point connectivity is required or surface re-structuring or re-sampling, resulting in a significant reduction of rendering time without compromising the quality.

Wimmer and Scheiblauer (2006) developed a rendering system for massive unstructured raw point clouds with a fast pre-processing based on Sequential Point Trees. By using this technique real time rendering (RT) of massive unstructured raw points became a more efficient way to manage the scan data, than the translation of the data into polygon meshes. In a more recent development Pintus et al. (2011) have proposed a RT rendering technique: 'Unlike previous techniques, our method renders in real time and on a frame-by-frame basis massive unstructured raw point clouds, solving both visibility and surface reconstruction with screen-space operators and without point attributes'.

Goswami et al. (2012) explored the idea of using hierarchical level of detail (LOD) and multi-way KD-trees, in order to simplify memory management. The hierarchical multi-resolution pre-processing and rendering of massive point clouds LOD using geo-morphing (continuous transitions between LOD's) and smooth point interpolation (smooth point interpolation through conventional blended splatting algorithms) achieves better rendering efficiency and allows for adjustment of resolution. As Goswami et al. (2012) have noted, PBR is gradually emerging as a viable alternative to polygon mesh rendering for very large geometrical models or set of data. Since for points mesh connectivity of the triangles is not required, making the preprocessing algorithms simpler, this also allows for and easy division of the renderings into several machines as not connectivity is required.

A different line of research proposes the combination of massive point clouds and meshes, through a conversion operation that replaces groups of coplanar points into polygons, eliminating a large number of points making the rendering more efficient. This hierarchical hybrid point-polygon structures approach was introduced by Gao et al. (2014), its advantage relies on the fact that for large planar surfaces polygons are more efficient that dense point-meshes, in that way the rendering load is reduced. Following a similar approach Kuder et al. (2013) have also suggested the idea of detecting continuous surfaces and replacing them with decimated meshes retaining color, textures and bumps from the original LIDAR data. These two later approaches are only efficient when flat surfaces are obvious and easy to determine, and as mentioned before there will be a loss of data during the translation process. In a similar line of research Canciani et al. (2013) have developed algorithms to extract CAD sections from point clouds of classical heritage buildings to rebuild the models as polygons from the sections or to generate 2D lines. All the mentioned conversion processes are time consuming, and data gets lost in the process, for that reason our research will focus on the opposite, basically in working exclusively with point clouds.

Currently there are available in the market several point-based software packages, that allow for visualization and in some cases also editing of the point clouds. Some of the software packages are more oriented to the pre-visualization of the point clouds obtained from laser scans and to export them as other formats, typically mesh formats, to be later used in other software packages. The translation of point clouds into meshes is not a straight forward task, as meshes tend to get heavy, and because the laser scans tend to have many empty areas, due to shadows, reflections and to areas difficult to reach during the scanning process. Below is a list of some of the most commonly used PBR rendering software:

CloudCompare, is a point cloud processing software that also works well with polygon meshes. It allows for rendering and visualization of point clouds and mesh models.

Leica Cyclone, is a software for the laser scanners, that facilitates the use of the scanners. It is made of several software packages for each specific task, such as merging and align several point clouds obtained from different scans runs of the same space. It can also model or edit point clouds, as well as exporting then into other the formats. It has a visualization engine and allows for measurements of the models.

MeshLab, is a software for editing non-structure 3D triangular meshes, that also works with point clouds. It allows for editing, cleaning, inspection, welding and visualization of the meshes.

Scene, is software created by 'FARO' scanners for the visualization of points-clouds obtained by its scanners. It is easy to use and allow for point and mesh visualization.

Gexcel JRC 3D Reconstructor, it is a software created by GEXCEL for exporting point scans to other CAD formats.

ToView, is a PBR software developed by the research team at the Department of Information Technology at A Coruna University. ToView is a visualizer implemented using C++, OpenGL for visualization purposes and QT, for the user interface. The visualizer has been implemented using OpenGL 4.3 and QT 5.3. ToView can efficiently handle point clouds of millions of points, due to the use of splats and level of Detail, LOD. ToView will be used, tested and evaluated later in the thesis, as a PBR methodology to bring closer 3D models from the 'real world', and 'artificially generated' models. Fig. 08, shows a raw example of a LIDAR scan, visualized as PBR with ToView.

Fig. 08. ToView visualization from a LIDAR scan using 'Faro Focus 3D', of a small historical urban environment.

2.1.6 Formats.

2.1.6.1 Polygon Meshes

Are a 3D format that defines 3D surfaces through a number of flat polygons. The most common type of meshes are polygonal and polyhedral meshes. Meshes are typically defined at several levels. The basic level are their 'vertex' or 'vertices', defined by its (X,Y,Z). In addition to the vertex the meshes contain 'edges', joining the vertices, which are vectors or vertices connectors, and defined by lines. Each group of closed edges, typically a triangle but can be any type of polygon, is what defines a 'face'. Each group of faces defines as 'polygon' which is a coplanar set of faces. Polygons will generate surfaces or a solid volume, also called smoothing groups. Each set of surfaces defines a group which it is basically a 3D object. Fig. 09, shows the wireframe of a polygon mesh of a 3D croissant. Mesh files also typically contain additional information such as color information, texture and the overall coordinates. Most meshes do this through its UV Coordinates, which are a 2D representation of the mesh where a 2D map, typically an image will be applied and scaled accordingly.

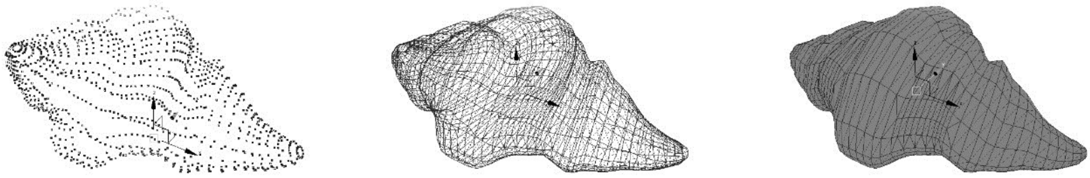


Fig. 09. From left to right, an example of 'vertex', 'edges' and 'faces', defining a 3D polygon of a 3D croissant.

A polygon mesh are points with connectivity information, through edges, creating faces typically as triangles, as it simplifies the files. It is important to note that curves cannot be directly made as polygon mesh format. Curvature is achieved in mesh formats by subdivisions the mesh into faceted straight edges or faces (Fig. 10).

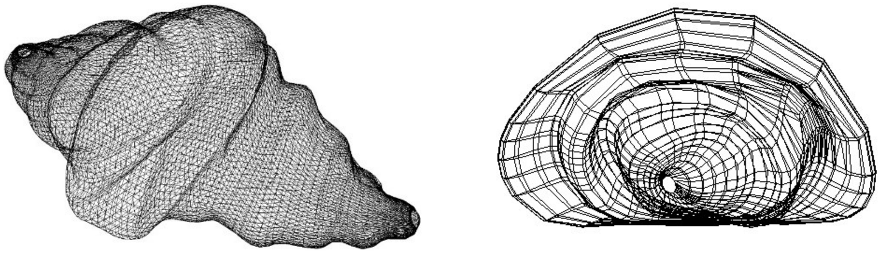
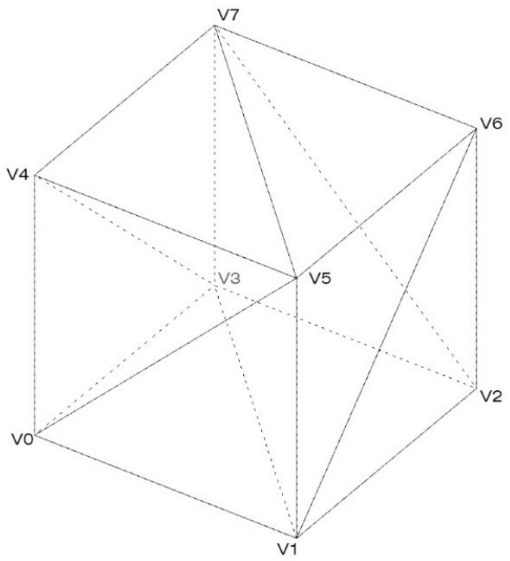


Fig. 10. Example of the wireframe of a polygon mesh of a croissant 3D model. Curvature is achieved through straight segments.

A number of short edges will achieve a curved smooth looking polygon, made out of straight surfaces, a faceted surface that at the right scale and number of polygons will look like a smooth one. Polygon meshes are the most common primitive used for architecture, design, animation and video games industries. There are several types of polygon meshes, the most common ones are the face-vertex meshes, winged-edge meshes, half-edge meshes, quad-edge meshes, corner-table meshes and the vertex-vertex meshes.

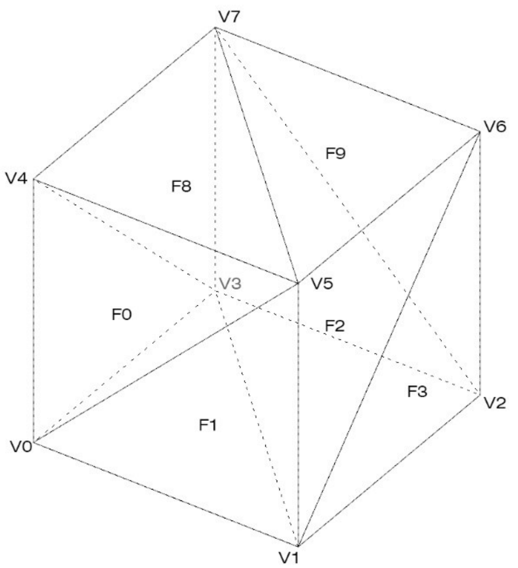


Vertex	Coordinates	Vertex List
V0	0,0,0	V1, V5, V4, V3
V1	1,0,0	V0, V3, V5, V6, V2
V2	1,1,0	V1, V3, V7, V6
V3	0,1,0	V0, V1, V2, V7, V4
V4	0,0,1	V0, V3, V5, V7
V5	1,0,1	V0, V1, V6, V7, V4
V6	1,1,1	V7, V5, V1, V2
V7	0,1,1	V4, V3, V5, V2, V6

Fig. 11. Example of a Vertex-Vertex (V-V) triangular polygon mesh of a cube with 8 vertices.

The simplest type of polygon mesh is the vertex-vertex mesh (V-V mesh). It represents an object as a set of vertices, and its connections to other vertices. Due to its simplicity is not commonly used, because do not contain vertex information it is not very efficient, Smith (2006). An example of a V-V mesh is shown in Fig. 11.
The most commonly used type of polygon mesh are the face-vertex meshes (Fig. 12). The represent both vertex and faces. This type of mesh is the one that most computer graphics software uses. This type of format is very simple and efficient; however, some complex operations are difficult to achieve with this type of mesh.

Another type of commonly used mesh are the winged-edge meshes, introduced by Baumgart (1975), they represent vertices, faces and edges of the mesh. This type of mesh contains more information, which allows for easier manipulation of the mesh, but also the files are larger and take longer time to process. This meshes are ideal for dynamic geometry, for animations. They won't be used in this thesis.



Vertex	Coordinates
V0	0,0,0
V0	1,0,0
V0	1,1,0
V0	0,1,0
V0	0,0,1
V0	1,0,1
V0	1,1,1
V0	0,1,1

Face	Vertex
F0	V0, V4, V5
F1	V0, V1, V5
F2	V1, V5, V6
F3	V1, V2, V6
F4	V2, V6, V7
F5	V2, V3, V7
F6	V3, V4, V8
F7	V0, V3, V4
F8	V4, V5, V7
F9	V5, V6, V7
F10	V0, V1, V3
F11	V1, V2, V3

Fig. 12. Example of a Face-Vertex triangular polygon mesh of a cube with 8 'vertices' and 12 'faces'.

2.1.6.2 NURBS.

Non-Uniform Rational Basis-Splines (NURBS), are a more recent type of 3D format, also known a rubber sheet geometry, which are topological geometries defined by algorithms. The curves and surfaces in NURBS geometries are based on splines, which are defined and manipulated by 'control points', 'weights' and 'knots', to define perfectly smooth lines and surfaces. NURBS have the advantage of being able to produce extremely complex curved geometry in a very efficient computational way. As Dunn (2012) has noted, before digital technologies, curved surfaces and forms were the product of approximations using tangents to circular arcs and straight line segments that were translated from the drawings to the buildings site.

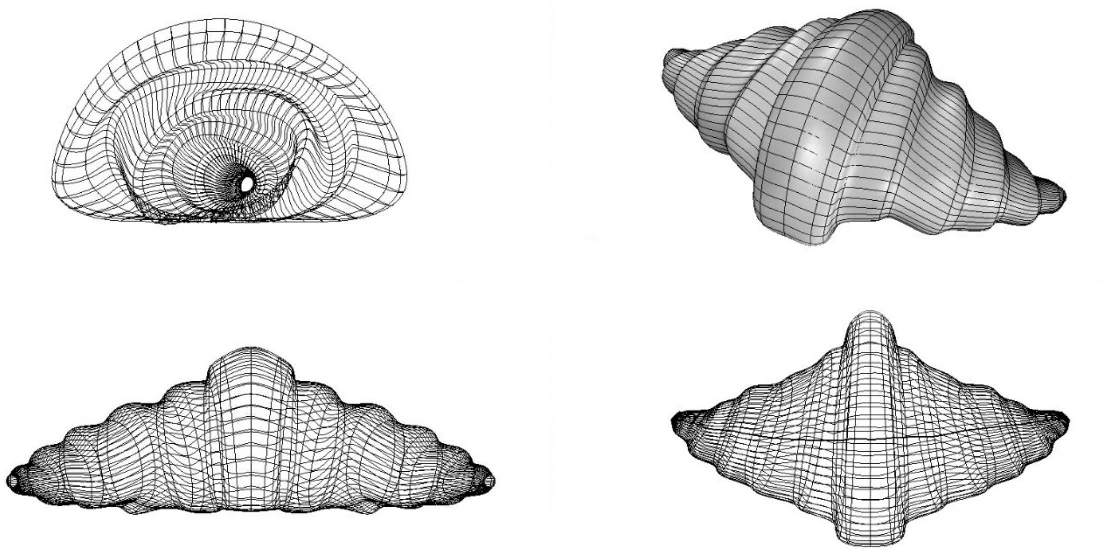


Fig. 13. Example of the wireframe of a NURBS of a croissant 3D model.

NURBS are mathematical representations of any curve or surface, from the most basic straight line to the most complex irregular surface, including organic shapes. The use of NURBS has many advantages. It defines its geometry through mathematical and very accurate geometrical equations, making the amount of information storage on a NURBS files smaller and more accurate that faceted primitives, such as polygon meshes. NURBS curves equations are defined by degree, control points, knots and evaluation rule.

The degree is a number, typically 1,2,3,4 or 5. Straight geometry is degree 1 or linear, circles degree 2 or quadratic, and free form curves range from degree 3 to 5 or cubic, but can be higher. The control points are a list of points which have an associated number called weight. Depending on the weight curves will be rational or non-rational. Curves such as circles, ellipses and other regulated geometries are rational, but most of curves drawn by following NURBS control points are non-rational, and called splines. The knots are a list of number defined by the number of control points, Fig. 13. The evaluation rule is the mathematical formula that will define each exact point of the curve based on its degree, control points and knots.

The evaluation rule will generate each point of the 'basis spline' through its mathematical formula. According to Piegl (1991) and Rogers (1991), the advantages of using NURBS are:

-Offer one common mathematical form for both, standard analytical shapes and free form shapes;

-Provide the flexibility to design a large variety of shapes;

-Can be evaluated reasonably fast by numerically stable and accurate algorithms;

-Are invariant under affine as well as perspective transformations;

-Are generalizations of non-rational B-splines and non-rational and rational Bezier curves and surfaces.

The mathematical formulation of a Basis-spline curve (Equation 01), is defined by the following polynomial function:

$$\mathrm{C(u)} = \sum_{i=0}^{n} N_{i,p}(u) P_i$$

Equation 01. Base polynomial definition of a Basis-spline.

Where 'P i' are the control points define by its coordinates (Xi, Yi, Zi), and 'u' is the knot vector (Figs. 14 and 15).

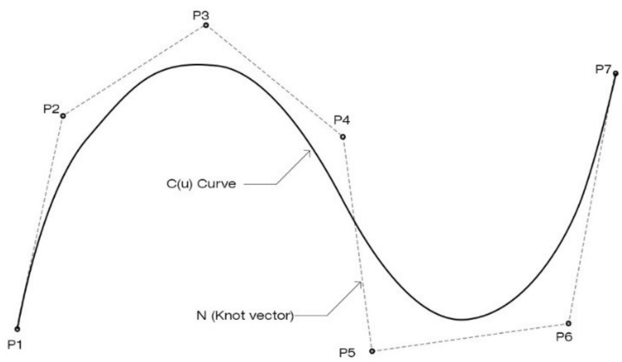


Fig. 14. An example of a NURBS Basis-spline, showing the control points 'P i', the knot vector 'N i,p' and the final resulting smooth curve 'C(u)'.

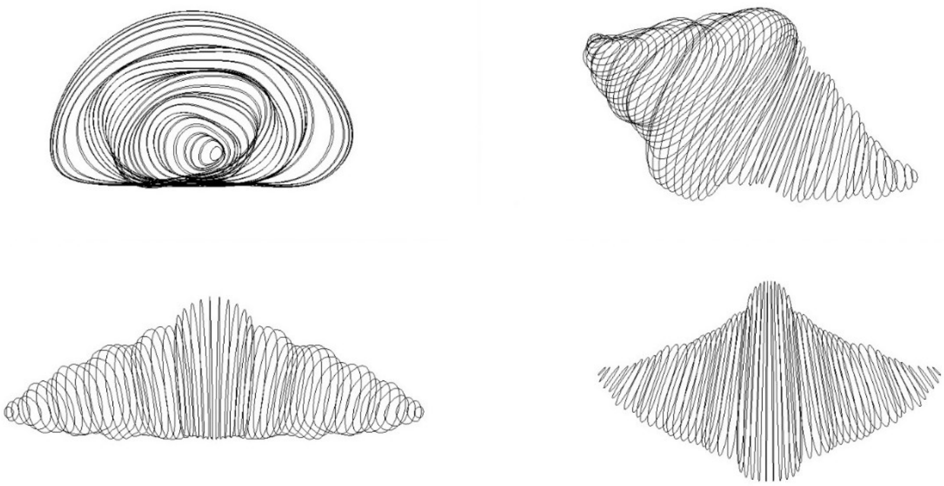


Fig. 15. A set of individual NURBS Basis-splines or C(u) used to generate a croissant 3D model.

The mathematical definition of a NURBS Curve is a vector-valued polynomial function (Equation 02), defined as, Piegl (1991), Piegl et al (1997):

$$\mathrm{C(u)} = \frac{1}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \sum_{i=0}^{n} N_{i,p}(u) w_i P_i$$

Equation 02. Base polynomial definition of a NURBS curve.

Where:

C(u): NURBS Curve

w_i : weights

Pi : control points (vector)

N_i,k : normalized B-spline basis functions of degree k

p: Curvature degree.

In a similar way that a NURBS curve is defined by it (u) vector, A NURBS surface is defined by its (u,v) vector coordinates (Equation 03), in a similar way, by combining ‘u’ and ‘v’ Basis-splines the following polynomial function is obtained:

$$\mathrm{p}(u,v) = \sum_{i=0}^{m} N_{i,p}(u)\left(\sum_{i=0}^{n} N_{j,q}(v)P_{ij}\right)$$

Equation 03. Base polynomial definition of a NURBS surface.

NURBS surface is define by its (u,v) coordinates, corresponding basically to two sets of basis-splines on two directions (Fig. 16). NURBS polynomial functions define splines or open single surfaces, but not solid objects. Multi surface objects are modelled in NURBS as a set of combined single surfaces, often referred as ‘poly-surfaces’. When a ‘poly-surface’ is closed, is referred as a solid.

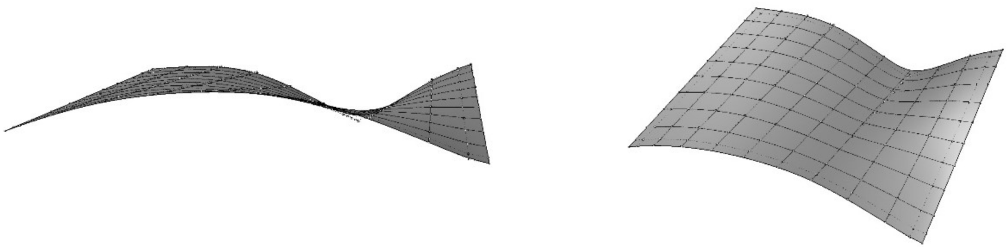


Fig. 16. An example of a NURBS Surface, showing the (u,v) coordinates and control points.

2.1.6.3 Point Clouds.

Also, known as unstructured point clouds are groups of unconnected points that range from just a few points to millions of them. This format is the simplest type of 3D format, as the points typically do not have connectivity information among them, point files are simpler than other formats. A point cloud contains point coordinates referred to an origin on a coordinate system. The vertices are positioned in the space by its coordinates (X,Y,Z). Sometimes the files also contain additional information, such as color (RGB), normals (tangents), luminosity, temperature, reflectivity, etc. The mathematical formulations and file formats are quite simple, however to represent a continuous line or a simple surface for instance, it is necessary to define a large number of points, making the datasets, often too large, compared to other formats (Fig. 17). Some of the most commonly used point formats are PTS, PTX, ASC, LAS, PLY, PCD and BPC.

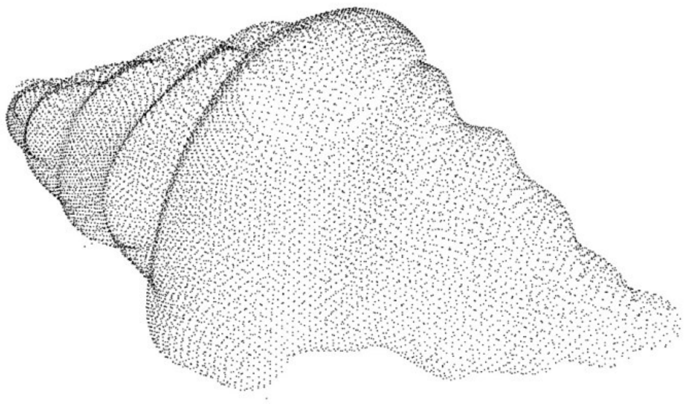


Fig. 17. Example of an unstructured point cloud of a croissant 3D model.

Typically, when using this format objects are represented by point samples of their exterior surfaces, being the interior of the objects hollow. Point samples are referred as splats, for rendering purposes. As mentioned in sub-chapter 2.1.4. unstructured point clouds are the common format used by laser 3D scanners, which capture them automatically and yields a dataset from real objects. As points are zero-dimensional, they do not have mass, so they do not have volume, area or length, which is the main reason why points are not commonly used for 3D applications, and why there are mainly used only as a reference entity to later generate lines, surfaces and volumes later. As explained in sub-chapter 4.1.5, PBR uses splats as rendering entities, which are small oriented surfaces applied to points, to simulate a continuous surface. Unstructured point clouds will be extensively used in the case studies as a novel way to merge 3D data obtained directly from the 'real world' with 'artificially generated' 3D data.

2.2 Generation of Complex Geometry.

2.2.1 Historical Background

Exploration of complex and non-orthogonal curved geometry has always been part of architecture and design. Geometrical complex curved elements were typically built based on traditions, and they replicated or improved previous experiences within the same time period and geographical area. Much of this complex geometry was purely for ornamental purposes, many times overlapping disciplines, such as sculpture and architecture. An ancient example among many other examples are the Caryatids, anthropomorphic columns, both decorative and structural. The Erechtheion Caryatides built in Athens in between 421 and 406 B.C. during Pericles period (Fig. 18).

Fig. 18. On the left, the Caryatids, at the Erechtheion Temple, fifth Century B.C., Athens Greece (chroniquesduvastemonde.com). On the right Agrippa's Roman Pantheon (Photo: Michael Vadon).

Some curved geometry had structural purposes, such as arches, vaults, and domes for instance. Complex curvature with structural function, such as Agrippa's Roman Pantheon, a semi-spherical Dome resulting of the rotational translation of an arch, built during the first Century B.C. (Fig.18).

As in the two previous examples, many cases of complex curved geometry had both a structural and an aesthetic function. A good example are late gothic Vaults, where originally simple structures, became increasingly complex for decorative purposes, such in the Fan Vaulting of King College in Cambridge (Fig. 19), completed in 1515. As Saltmarsh (2015) noted the Chapel's celebrated fan vaulting ascends to the elaborate bosses punctuating the great spine of the building, designed by the Master Mason Simon Clerk, a structurally performing geometry becomes an ornamental motive, blending function and ornament.

Fig. 19. King's College Chapel Fan Vault, Cambridge, UK. Completed in 1515. (Photo: Seiser+seiser).

Vaults and Domes were build based on traditions and historical continuity by Stone Masons, which were both craftsmen and designers. Elaborated facades were built and designed until them, as sculpture and architecture hybrids. These complex geometrical structures were made without or with a very limited amount of drawings. The Latin term 'architectus' has a Greek origin, and means 'Chief builder', but it is not until the late renaissance, around the mid sixteenth century, that the term 'architect' starts to be used, when also paper and pencils start to be used in the design of buildings, Pacey (2007). During the Renaissance period developments in architectural representation had a tremendous impact on architectural form and geometry, freeing the Master Masons from the routine of reproducing previously built examples.

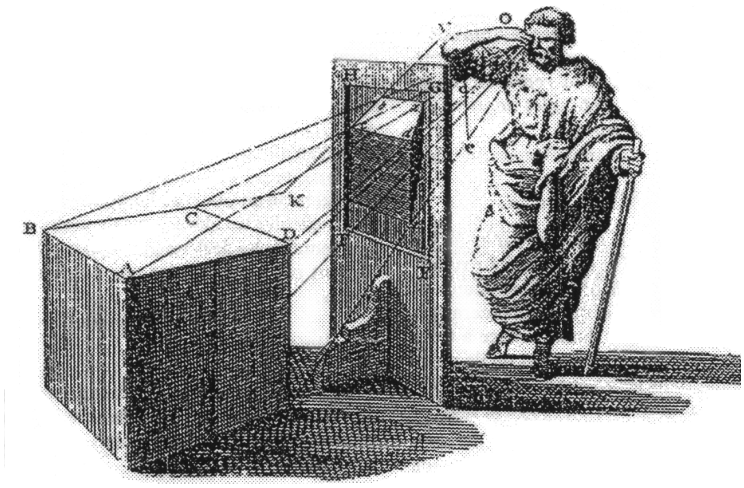


Fig. 20. Alberti's Visual Pyramid, from his book 'Della Pittura' 1435. Alberti (1435).

The father of modern visualization was Leone Battista Alberti, he developed the basic rules of linear perspective, Vasari (1570) explained the relevance of Alberti's findings on his book 'The lives of the most eminent painters' (Fig. 20). The first of the three 'Books of Arts, 'Della Pittura' written by Alberti (1435) and its Latin version 'De Pictura' Alberti (1441), were not only the precursors of the linear perspective, but also set the basis for today's graphical 3D projections on modern computer screens, as Kittler (2001) has noted regarding Alberti's discovery: 'Right angles become oblique, equal distances become unequal, and parallel lines become bundles of rays from an infinitely distant vanishing point, until the finished painting translates all three spatial dimensions into a linear-perspectival illusion'.

Alberti's findings were further developed by Dürer (1525) on his book 'The Painter's Manual', where shows a method to draw perfect perspectives from real objects (Fig. 21). His artifact generated points, by using ropes to ray trace points from a real object into a fixed spot, to then intersect them with a mobile plane. Not only the Dürer artifact will generate a sequence of points to later being manually interpolated into a line, but basically this machine set the basis of what a modern 3D computer rendering engine does.

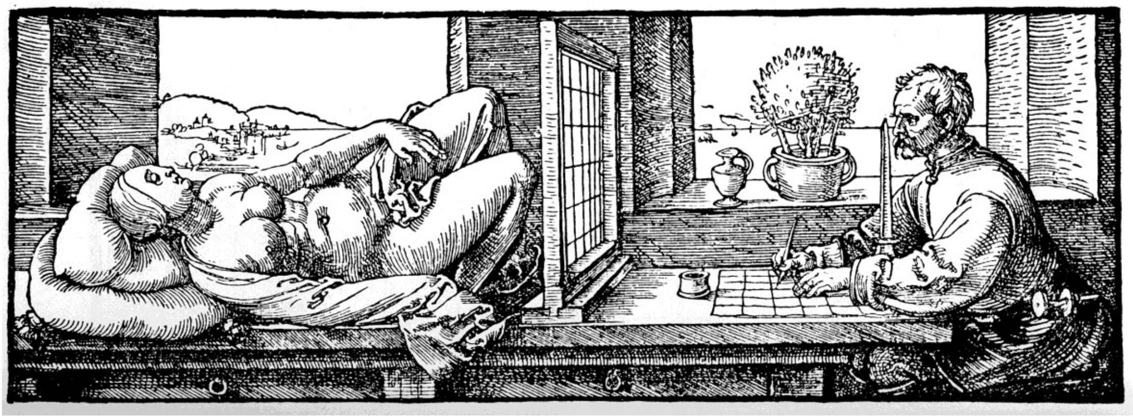


Fig. 21. Illustration from Albert Dürer's 'The Painter's Manual'. Strauss (1977).

Dürer didn't intend to draw points, but to interpolate points to replicate edges, contours and surface intersections, to obtain a 2D image in the same way as the human eye does. Kittler (2001) also compares the perspective construction machine of Dürer with modern 3D algorithms: 'What Dürer begins to at once write and draw up as a perspectival construction is something that we today are more familiar with than his contemporaries. It is the Europeanized name of a great Arabic mathematician, i.e. an algorithm'. As Kitler noted on his remarks regarding Dürer, the term 'algorithm' has its origin in the word 'Algoritmi', which is the Latinized name of Persian mathematician Al-Khwarizmi, 825AD., Anglin (1994).

The idea of PBR basically doesn’t differ much from what the Dürer artifact did. If Dürer had run his machine to produce thousands of points, his machine would have indeed been the first point-based rendering device (Fig. 22).
The discoveries of Alberti and further developments by Dürer, will set the rules of three dimensional representations for many centuries to come, and even set the basis of current 3D computer rendering. Further developments in perspective and orthogonal projections took place over the next few centuries, but they didn’t differ much from those original findings of Alberti and Dürer, until the development of Descriptive Geometry.

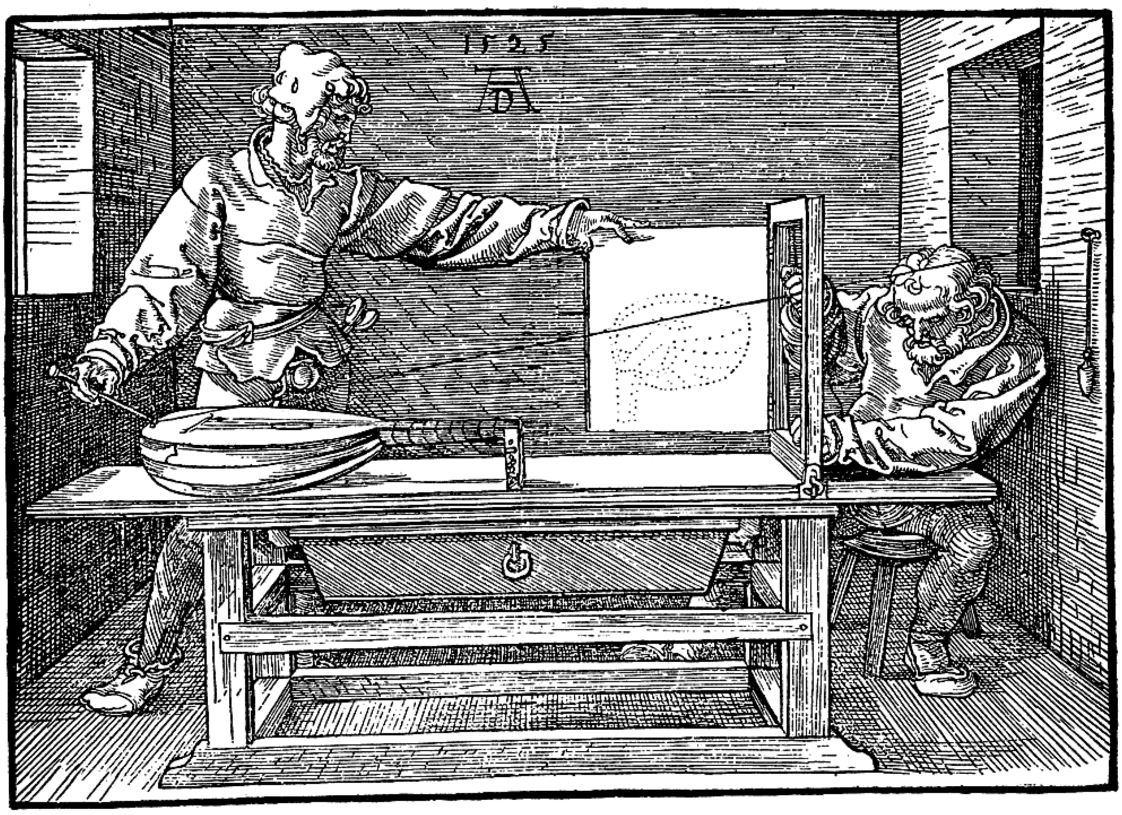


Fig. 22. Man, Drawing a Lute, a woodcut by Albrecht Dürer. In the 1525 edition of *this manual, Dürer shows an apparatus to create a perspective drawings. Strauss (1977).*

Following the Renaissance period new experiments that departed from the previous knowledge transmitted generation through generation started to take place, by returning and reinterpreting the classic architecture of Greece and Rome, and eventually reinventing the classic architecture. Andrea Palladio started diverting from the ancient classical styles and initiates an experimentation process that eventually culminates with Francesco Borromini, starting a new era of complex formal architecture today known as Baroque. Borromini's San Carlo Alle Quattro Fontane design (Fig. 23), completed in 1646 was a direct consequence of the development of architectural representation that took place two centuries before. Borromini's geometry even if visually complex followed well known geometrical shapes, such as ovals, circles and ellipses, Hill (2013).

Fig. 23. Francesco Borromini plan (early 1660's, Albertina museum), and picture of San Carlo Alle Quattro Fontane, Rome, completed in 1646. Bellini (2004).

Once the formal and ornamental excesses of the Baroque period came to an end, the classic rigor took over again during the Illustration in post-revolutionary France, on what has been called Neoclassicism. As Rocker (2010) has noted mathematician Gaspard Monge (1799) radicalized geometry by developing a system of representation of complex geometry based on orthogonal projections, combining mathematical calculations and visualization, in order to allow for orthographic representation of complex curved trajectories, setting the based for modern visualization of complex surfaces and the orthogonal projection system, which is what today are commonly known by architects as plans, sections and elevations. In Monge (1799) own words: ‘Descriptive geometry has two objects: the first is to stablish methods on drawing paper which has only two dimensions, namely length and width, all solids from nature which have three dimensions, length, width and depth, provide however, that these solids are capable of rigorous definition. The second object is to furnish means to recognize accordingly an exact description of the forms of solids and to derive thereby all truths which result from their forms and their perspective positions’. Since then architects had the knowledge and tools to represent complex geometry, Monge and Durand findings became the standard representational methodology, and manual drafting on paper remained the main method of architectural representation, until the very end of the twentieth century.

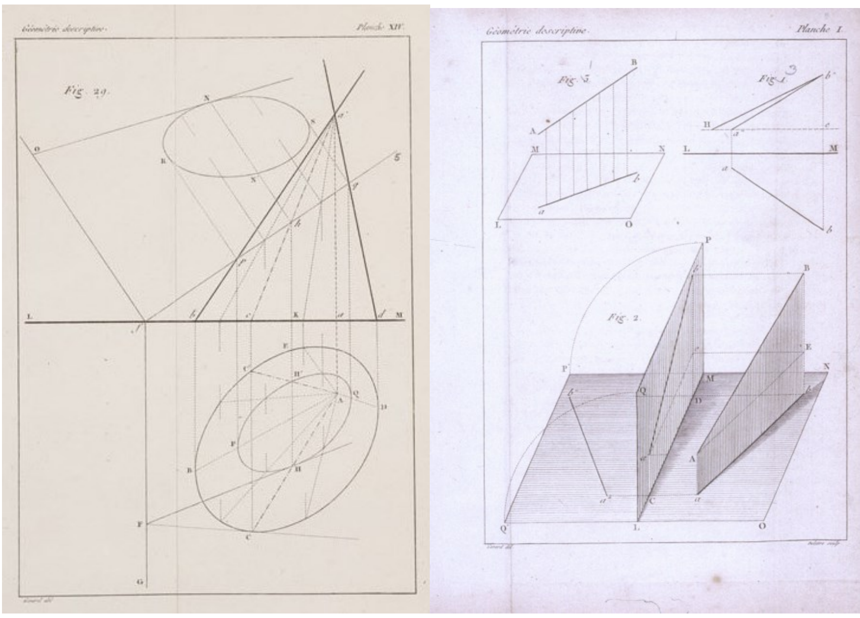


Fig. 24. Illustrations from Monge’s ‘Descriptive geometry’ Book. Swetz and Katz (2011).

The development of descriptive geometry by Monge (Fig. 24), gave architects representational tools to accurately manipulate and visualize even more complex geometrical shapes, opening new possibilities for the exploration of geometrical complexity. Drawings as a codified method of representation were not standardized until the beginning of the nineteenth century when Durand (1805) based on Monge findings develops a system of orthogonal projections based on a Cartesian grid, Rocker (2010). Durand an important figure in the Neo-classicism, developed a system of simple, modular elements for architecture, anticipating the standardization in architecture that will happen during the twentieth century.

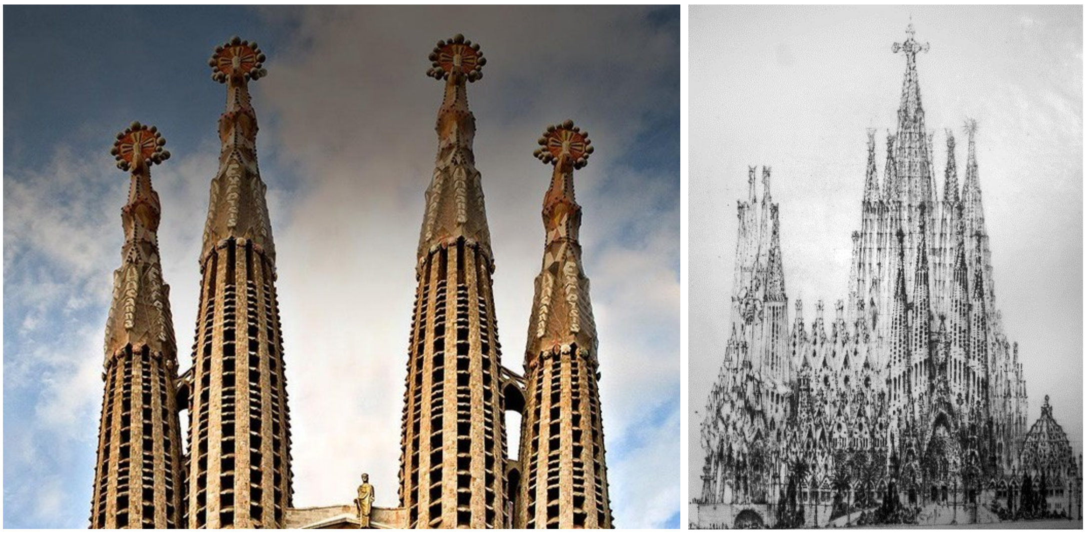


Fig. 25. Gaudi's Sagrada Familia in Barcelona. Nativity facade completed in 1936. Interpretative drawing by Lluis Bonet I Gari (1945). Davis (2013).

The evolution of geometrical representation gave birth to a large variety of historical styles during the nineteenth century, by further freeing architects through representational developments. By the late nineteenth century, architecture starts diverting from the styles of the past, in response to technological and representational advances. The rupture culminates with the Art Noveau, Modernism and Art Deco styles, all highly formal with complex curvature. The advances in architectural representation allowed for those highly formal new styles, by giving the architects the necessary tools to accurately represent geometry never tested before.

An example among many others are the work of Antoni Gaudi (Fig. 25), and Victor Horta. As Oxman (2010) has noted regarding the Sagrada Familia by Antoni Gaudi, the process currently undergoing to complete the temple is associated with parametric design systems, Burry (1999).

During the twentieth century, the introduction in architecture of new materials and technologies, resulted into a radical transformation of architecture. The discipline reinvents itself in a radical rupture with the formalistic past, initiating the 'Modern Movement'. A period of modern abstraction and formal rigor took off in the 1930's lead by Le Corbusier and Mies Van Der Rhoe, and continued during the postwar period producing an architecture discourse based in formal austerity. Despite formal functionalism, curved geometry prevailed as a recurrent motive in some works, such as in Le Corbusier's 'Notre Dame du Ronchamp', just to name one of the numerous examples.

Fig. 26. Picture of Utzon Sidney's Opera House, Sidney. Completed in 1973. (Photo: Adam J.W.C.).

During the post-war year's architecture stars embracing more extreme geometrical challenges, but always within the boundaries of Cartesian descriptive geometry. An example is the iconic Utzon's Sidney Opera House completed in 1973 (Fig. 26), Its geometrically complex looking roof was initially conceived as paraboloids, but due to construction and budget constrains it was materialized as sectional triangular spheres, easily described by descriptive geometry. (Fig. 27).

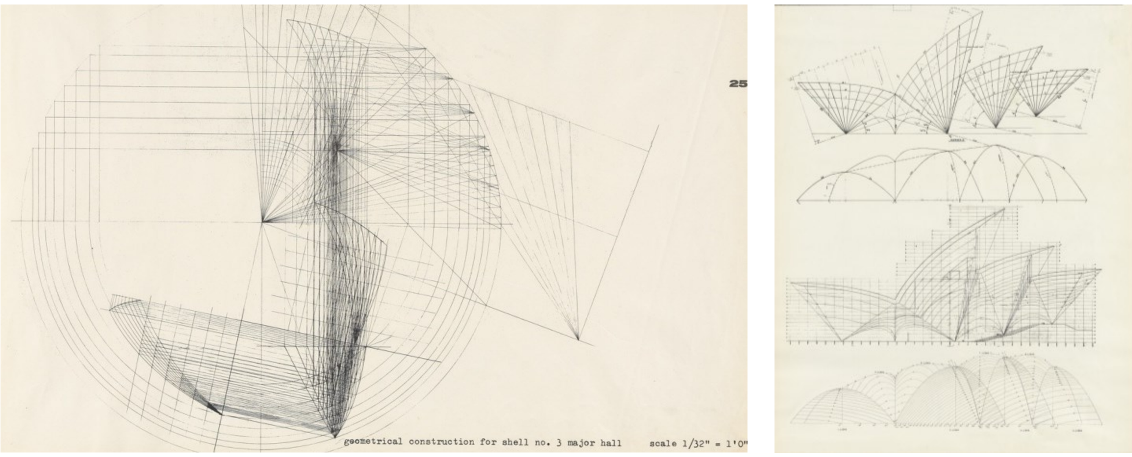


Fig. 27. Drawing from the 'Yellow Book' of Jorn Utzon Sidney's Opera House, 1962. Showing the geometry of the exterior shells. Utzon (1962).

The first attempt on using a computer for drawing purposes was made by Ivan Sutherland (1963), he devolved an early computer program called 'Sketchpad' as his Ph.D. thesis, also known as 'Robot Draftsman' is considered as the pioneer of Computer-Aided Design (CAD) and Computer Graphics (CG) programs. During the 1960's and 1970's CAD programs were developed and by the 1980's started being used by architects. As Dunn (2012) has noted: 'Even as recently as the early 1990s very few architectural practices used computer other than as a time saving devices for administrative task. The introduction of the paperless studios at Columbia University in an attempt to integrate computers into the architectural design process was seen at the time as a novel distraction rather than a serious proposition of the future of architectural design'.

By the late 1990's as computation gained relevance to design and the use of architectural software became a common practice. The main purpose of CAD programs was to produce two dimensional drawings by mimicking traditional drawing techniques and standards into a computer program, basically to computerize the findings and technique proposed by Monge and Duran in the nineteenth century. The programs were designed to draw lines, simple polygons, circles, splines, text, dimensions, hatches, etc. very much in the same way that it was done previously manually. The use of personal computers in architecture was basically limited to the production of line-based technical drawings to be later printed into a plotter. By using a computer, the information would be digitally stored, and will allow for much easier to revisions, copies and reproductions. CAD software packages were also particularly efficient to make modifications on existing drawings, as the programs allowed for local revisions without the need to redraw the drawing all over again. As Iwamoto (2009) has noted: 'For many years, as the process of making drawings steadily shifted from being analog to digital, the design of buildings didn't reflect that change. CAD replaced drawings with a parallel rule and a lead pointer, but buildings looked pretty much the same, one form of two-dimensional representation simply replaced another.'

By the late 1990's architects started turning their heads to other software packages initially designed for 3D graphics, for the gaming, animation and motion picture industry. These programs were 3D based, and their purpose was to replicate reality, by generating animated perspectives from a given Cartesian 3D geometry, using basically similar rules than the Dürer machine. As computational capabilities grew, the programs were also able to replicate the real world on a 3D model and then to animate still frames by using cameras and a timeline. By the beginning of the twenty-first century, not only the use of some of those 3D software's was widely spread in the architecture industry, but also software packages exclusively designed for architecture started to introduce those 3D capabilities, with the main purpose of designing directly in 3D, allowing architectural representation to depart from the Euclidean geometry, as Dunn (2012) has noted.

When 3D modeling and rendering was fully introduced in architectural design, the shift from lines to three-dimensional polygon meshes didn't happen right away, and the traditional lines used for technical drawings as a mean of representation are still widely in used nowadays. New terms such as Digital Design, Digital Fabrication, 3D Printing and Computer Graphics were incorporated to the semantics of the architectural discourse.

Fig. 28. Frank Gehry's Gugenheim Museum, Bilbao, Spain, 1997 *(Photo: Edwin Poon).*

As a direct result of the introduction of computers, architecture has witnessed in an increasing number of buildings that have embraced extremely complex curved geometry, which supposes a rupture with previous formal historical precedents. This new types of geometries were just impossible to be built and designed without the assistance of computer technologies. The completion of the first large scale building with such a formal language was the Bilbao Gugenheim Museum by Frank Gehry completed in 1997 (Fig. 28).

It was soon to be followed for by a series of geometrically complex buildings worldwide, which embraced asymmetry, irregularity and non-regulated geometries. Utzon's Sidney Opera House iconic spheres were replaced by irregular geometry in Gehry's Museum, none of the exterior and interior surfaces follows a curve easily defined by regulated descriptive geometry. The building however was not designed by a computer, but through physical paper models. Gehry was aware of the capabilities of new software packages for the aeronautical industry, in particular the software 'CATIA', so he designed an irregular building in paper conscious that the new computer technologies will allow for its fabrication. The models were manually scanned, and redrawn into 'CATIA', to be later fabricated, through CAD technology, becoming the largest digitally fabricated building at the time of its completion.

It can be argued that in the same way that the discovery of the perspective by Alberti changed architectural history paving its way to the renaissance and eventually the baroque, CAD and 3D technologies have had similar impact in recent years. The developments on 3D software have led to a geometrical reformulation of architectural design. The first and most obvious manifestation have been new generation of buildings of extreme complex curved geometry, that simply weren't possible to build before, without the aid of computation. Further developments in NURBS and digital design had led to a second generation of buildings, this time both digitally designed and fabricated. This new family of complex curvature buildings follow a fluid continuous geometry generated by NURBS modelling, such as the work of architect Zaha Hadid, just to name one of the first and most notorious examples using this type of geometrical solutions, (Fig. 29).

Fig. 29. Zaha Hadid's Heydar Aliyev Center in Baku, Azerbaijan, 2012. (Photo: Zaha Hadid Architects).

The geometry of Zaha's latest generation of buildings was directly influenced by the adoption in Zaha Hadid's studio of the software Rhinoceros, that led to a geometrical reformulation of her design. Three Dimensional methodologies have not only been a helpful tool for digitally designing but have generated a new design typologies themselves. Fig. 30, shows the construction process of the 'Heydar Aliyev Center', basically a physical materialization of a NURBS surface, directly extracted form a 3D model.

Fig. 30. Construction picture of the Heydar Aliyev Center in Baku, Azerbaijan, completed in 2012. (Photo: Zaha Hadid Architects).

Among many other notorious example of NURBS driven geometrical aesthetics, is

‘The City of the Culture of Galicia’, in Spain, designed by New York architect Peter Eisenman, (Fig. 31).

Fig. 31. Status of Peter Eisenman’s City of the Culture of Galicia. (Photo: Howard Kingsnorth).

The buildings were designing using physical models and Rhinoceros NURBS modelling software. The NURBS software drove the geometrical formulation of the smooth curvatures, both on the interior and exterior of the building. It can be said, that the software has not only influenced, but formally driven the curvature of the design. Fig. 32, shows a NURBS model of the interior ceilings, physical paper model built directly from NURBS data.

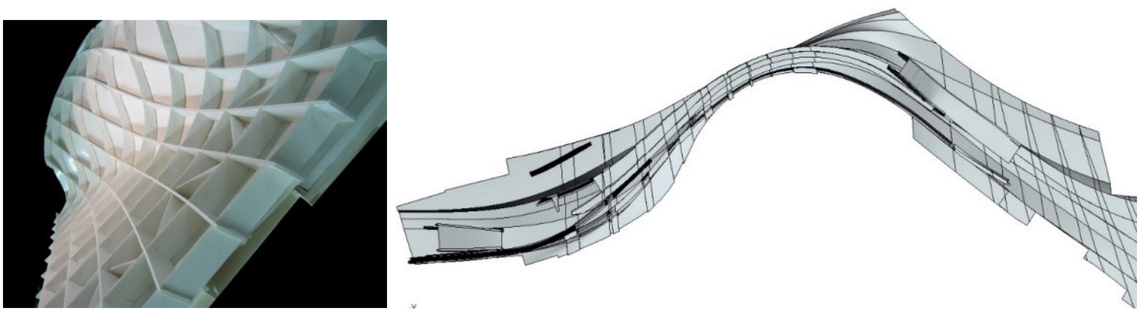


Fig. 32. City of the Culture of Galicia: Interior ceilings. (Models by Author).

The use of NURBS geometry has in recent years being also applied to high-rise buildings. This process of application of complex curvature to high-rise structures is particularly interesting as high-rise buildings are typically just vertical extrusions of a grid, to perform well to gravity loads and lateral forces. The first generation of NURBS-based high-rise structures, have kept rectangular floor plan layouts to maximize the interior efficiency of the building, but have developed complex 3D geometry facades, using NURBS-based decorative surfaces, such as the 'Aqua Tower' in Chicago by Jeanne Gang completed in 2009, or Frank Gehry's 'Beekman Tower' in New York, completed in 2010. (Fig. 33).

Fig. 33. From left to right: The 'Aqua Tower' in Chicago, 2009 (Photo: Studio Gang). Frank Gehry's 'Beekman Tower' New York, 2010 (Photo: Manolo Franco). The 'Shanghai Center Tower' super-tower, 2014 (Photo: Gensler).

The application of curved geometry in some other cases had a performative function, such as reduce wind lateral forces, or to find structurally stable geometries. Just to name one of the many examples the 'Shanghai Center Tower' super-tower, completed by Gensler in 2014 and toping 632m (Fig. 33 right) which uses curved geometry to effectively reduce wind loads.

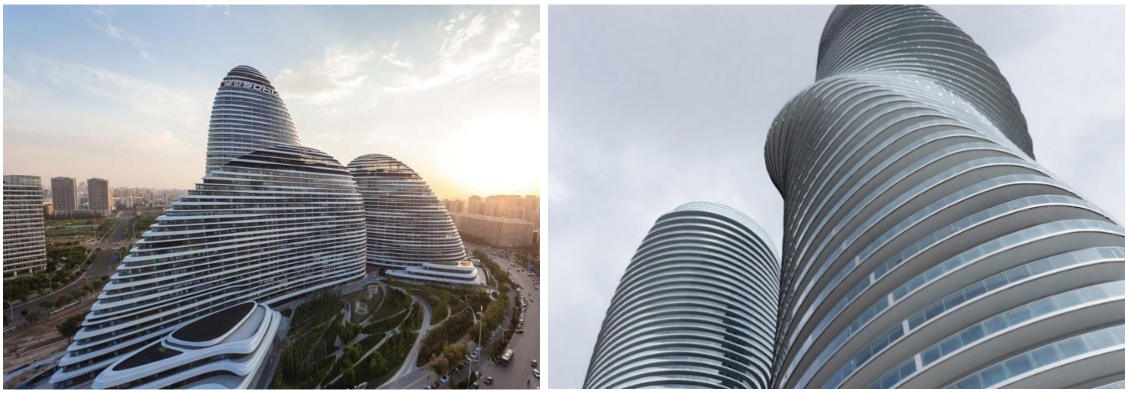


Fig. 34. On the left Zaha Hadid's Wangjing SOHO, 2014 (Photo: Zaha Hadid Architects). On the right MAD's Absolute World Towers in Toronto, 2012 (Photo: Iwaan Ban).

Other are more formal examples of high-rise's that use curved geometry as an aesthetic motive, for the overall massing of the building. Some examples of this type of architecture are shown on Fig. 34.

In addition to NURBS-based geometrically based buildings, additional more complex computation processes under development have the potential to reformulate again the whole basis of architectural design.

Parametric architecture, which proposes a design methodology based on the generation of design routines (codes) and the manipulation of its parameters. Algorithm architecture that proposes a methodology based in the introduction of algorithms in the design process. Furthermore, processes of AI and evolutionary morphogenesis are being slowly introduced and tested in architecture. The next sub-chapters will introduce those concepts, which are one of the main objectives and research topics of the thesis.

2.2.2 Parametric and Algorithm Design.

Computers have not only made possible the manipulation and direct manufacturing of forms and designs unthinkable before, but also have opened new ways of experimentation in design. Recently, architects have started to be interested in the possibility of using computational tools to self-generate designs. The idea is the use of parametric relations, algorithms and self-organization processes to create designs with limited human interaction.

The idea of parametric design was first proposed by Sutherland (1963), who also created the first computer graphics parametric interface, and predicted the current trends. In Timothy Johnson's video at the Lincoln Laboratory (1964) demonstration of Ivan Sutherland's (1963) Sketchpad the lines are joined by parametric relationships. Per Johnson's description in the video: '…the computer understands the geometry of the drawing in here… if I point to this particular point and tell the computer to move that point, it will move not only that point, but all the three lines attached to it'.

Parametric Design explores the idea of creating sets of rules through parametric relationships and algorithms that will evolve an original design, by the manual manipulation of its parameters by the user, obtaining results not necessarily expected by the designer. Per Dunn (2012) parametric design enables the designer to define relationships between elements or groups of elements and to assign values or expressions to organize and to control those definitions. It is usually applied with a three-dimensional CAD program.

Per Davis (2014) another observable characteristic of a parametric modelling, besides the presence of parameters, is that the geometry changes when the parameters change, leading to the claim that change is parametric. Yessios (2013), founder and CEO of the modelling software 'FormZ', summarizes the history of this interpretation: 'Initially, a parametric definition was simply a mathematical formula that required values to be substituted for a few parameters in order to generate variations from within a family of entities. Today it is used to imply that the entity once generated can easily be changed'.

Parametric design creates connections and relationship in between all the different elements of a design. By manipulating one element of the design, other elements will change automatically to adapt to the change, typically through an automatic change of parameters or values which are related, as in an equation system. Per Weisberg (2008), parametric modelling is 'more similar to programming than to conventional design'. Davis (2013), explicitly explains how parametric modelling through codes and software engineering share the similar foundation and principles: 'software engineers creating computer programs and architects designing with parametric models share similar challenges, which can often be addressed with similar research methods and similar design practices'.

To illustrate this with a simple example, a wall is typically represented by a set of parallel and perpendicular lines in an architectural drawing that define the outline of its geometry, however in parametric design a wall will become an object with values associated to it, such as its thickness and height, when changing the values of thickness and height by manually manipulation the parameters numeric values the walls will update themselves on all the drawings based on the input values. When this logic is applied to more complex systems or geometries the manipulation of the parameters has the potential to become a design tool itself. For example, by generating a surface of complex curvature, controlled by some predefined values, when manipulating the parameters, the surface will automatically change its shape. The advantage of parametric design is that once the parametric relationship has been established and codes developed, the system runs by itself in a very fast and efficient way. The disadvantage of this design methodology is that developing the parametric codes is a very time consuming process, that many times is not time efficient. Nowadays many codes can be downloaded, in that way codes can be shared by designers, each code is made to generate a specific type of routine. In his book 'Elements of Parametric Design' Woodbury (2010) explains: 'Parametric design depends on defining relationships and the willingness (and ability) of the designer to consider the relationship-definition phase as an integral part of a boarder design process. This process of relationship creation requires a formal notation and introduces additional concepts that have not previously been considered as part of design thinking'.

Parametric design is not by itself more successful than traditional design methodologies, it is just a different approach, on which a design routine is programmed to automatically to generate different design options based on the modification of numerical values, on what Burry (2003) has defined as: designing the design. More recent approaches involve generative design using generative algorithms, taking advantage of the analytical potential of computers to deal with the inherent limitations of human beings, Marcos (2010). The idea is to use generative algorithms to define formal structures created from a script. This approach according to Dunn (2012) requires a total shift in the way use the computer, instead of an interface to facilitate task to individuals the computer becomes a tool of generation of ideas, through a way of thinking that humans are not capable of. According to Terzidis (2006): 'Computerization is about automation, mechanization, digitization, and conversion. Generally, it involves the digitization of entities or processes that are preconceived, predetermined, and well defined. In contrast, computation is about the exploration of the indeterminate, vague, unclear, and often ill-defined process; because of its exploratory nature, computation aims at emulating or extending the human intellect. It is about rationalization, reasoning, logic algorithm, deduction, induction, extrapolation, exploration, and estimation. In its manifold implications, it involves problem solving, mental structures, cognition, simulation, and rule-based intelligence, to name a few'.

Algorithm architecture allows the designers to access the computational processing power of the computer. By using this methodology new explorations that the designers were not initially aware of might open, leading to previously unknown solutions.

2.2.3 Morphogenesis

Morphogenesis is the evolutionary development of an organism's form or part. The morphology of living organisms has slowly evolved over time into complex forms. Such transformations can also be simulated by computational processes.

The term of Morphogenesis was used by Alan Turing (1952) on his article titled 'The Chemical Basis of Morphogenesis', on which he explored the recurring numerical patterns of flowers, demonstrating with mathematical tools how flowers generated their complex geometry by self-organizing processes. Turing as a WWII code breaker and one of the fathers of modern computers had focused on pattern recognition within an apparent chaos. Morphogenesis focuses on bottom-up logic of form finding emphasizing performance over appearance, Leach (2009).

The term morphogenesis, applies to biological growth, but also to the auto-generation of artificial geometries, such as 3D models. It is important to note the difference between manually generating a geometry, like by sculpting, or manually model a 3D geometry, to the auto-generation of geometries through growth and evolution, as it happens in biological systems. The generation of open forms through the use of scripts, as Delanda (2005) has noted, should be defined as topological instead of geometrical, as it is the relation of its parts that defines the whole. Closed forms are defined geometrically, however open form does not necessarily fit on the geometrical space. Morphogenesis can be achieved through embryo genetic processes, through replication and evolutionary elimination, Andersen and Salomon (2012). Biological growth is a selective system, which contains generators of diversity, such as mutations or synaptic changes that are unpredictable. Topological systems such as point clouds seem more ideal for genetic growth than non-topological geometries, as they are more closely related to the principles of self-organized systems, and bottom up logic, as the cells in an organism, ant colonies or the binary elements of a computer, as Johnson (2001) has noted. Algorithm evolutionary design allows for design explorations based on producing further generations of a design. The idea is to let the design evolve in a similar way genetics do. By taking advantage of computational technologies, and the principles of natural selection computers can be used to seek the best or the most appealing solution to a 3D geometry. The basis of evolutionary morphogenesis through genetic algorithms, will be further explained on sub-chapter 2.3.

2.2.4 Self-Organization

Morphogenesis and adaptive self-organization are closely related terms. Morphogenesis is based on the form generation of an individual or object over time, on the other hand self-organization refers to collective behavior, when a group of individuals, or objects through simple rules of interaction generate a complex collective behavior resulting into a final structure that evolves over time and behaves as one entity. In self-organization, complex patterns can emerge from very simple rules, by the local interactions of simple unconnected elements to create complex self-organized systems, a process also called emergence (Johnson 2001). Emergent behavior is a type of self-organization on which simple components with simple rules of interaction evolve arranging themselves into a form. Form and evolution is not linear as behavior changes form evolves in different ways, and extreme complexity is reached through simple behavior.

Some biological examples of adaptive self-organized behavior are ant colonies, and bird's flocks, where many individuals making individual simple decisions create very complex communal behavior, leading to higher level collective behavior. The analogies can be applied to architecture as well, as Kolarevic (2003) has noted: ‘The emphasis shifts from the making form to the finding form, which various digitally-based generative techniques seem to bring about intentionality. In the realm of the form the stable is replaced by the variable, singularity by multiplicity’. Weinstock and Stathopoulos (2006) have noted: ‘In biological systems self-organization is defined as a process in which a pattern at the global level of a system emerges as a result of interactions among the lower-level components of a system. In artificial systems, interactions can be specified as rules which drive the lower-level components. These are executed by local information which is not related to the global pattern. Complexity increases when integration and differentiation increase.’

Among human made structures, the most remarkable example of adaptive self-organization is the organization and growth of cities. The emergence and evolution of cities, responds to the similar self-organizing principles of a morphogenesis processes.

2.2.5 Simulation of Urban Growth

Many authors have noted links between urban growth and biological growth such as pioneering studies by Weaver (1958) in the Natural Sciences. Jacobs (1961), who applied Weaver’s notion of organized complexity, further theorized that although cities are a model of disorder they somehow work successfully. Johnson (2001) has claimed that the growth of large urban metropolitan areas resembled the growth of biological organisms. A similar analogy was made by Al-Sayed and Turner (2012) who described cities as complex organisms in constant evolution, its growth governed by both an evolutionary process and a self-organization process.

Al-Sayed and Turner (2012) also noted how most of the computers models that simulate urban growth, are developed based on mechanisms that characterize biological systems, rather than spatial systems.

The main precedence for the use of computer models to simulate urban growth is Cellular Automata (CA) modelling. According to Leao et al. (2004), since the 1980s and the introduction of computation and self-organization, the traditional top-down approaches for the simulation of urban growth have shifted towards bottom-up approaches. Clarke et al. (1997) developed the first CA-based SLEUTH model, which was applied with satisfactory results in the simulation of urban growth for the San Francisco Bay area Clarke and Gaydos (1998). Other CA models were developed by Wu (1998) to analyze forms of city growth for different development strategies, and Li and Yeh (2000) for sustainable development of agricultural land. In a different study, Li and Yeh (2002) combined neuronal networks and CA using GIS to simulate the evolution of multiple land uses. Leao et al. (2004) further noted how urban growth has generally been the product of individuals making decisions within existing regulations even if these decisions did not necessarily follow the most optimal patterns. Due to this, CA models have also been used in combination with other techniques. White and Engelen (1997) included a stochastic factor into their CA models while Clarke and Gaydos (1998) opted to use probabilistic random processes. Rienow et al. (2014), used CA-based SLEUTH spatial modelling for simulating urban growth in combination with multi-agent systems (MAS), which are well-suited to capture individual decision making. Some other studies have used genetic algorithms to calibrate the CA modelling for urban simulation, Li et al. (2007) and landscape metrics Li et al. (2013). Other recent studies combined CA with swarm intelligence-based algorithms Delavar et al. (2016).

Outside of the examples listed above which mostly used cellular automata models as a base, no other attempts to produce similar types of studies using exclusively genetic algorithms for simulating urban growth have been found. Since urban growth is the result of a collective and largely unplanned effort that does not necessarily follow the most logical, advantageous nor efficient patterns, the use of artificial intelligence, and more in particular evolutionary computation seems to present an ideal and novel approach to simulate vertical growth.

2.3 Artificial Intelligence: Evolutionary Computation.

2.3.1 Fundamentals.

Artificial intelligence is the science that creates programs for machines to imitate human behavior. The first developments date back to the 'Turing Test', (Turing 1937 & 1950), when the question if machines could think was posed, he called it the 'imitation game'. Turing estimated that within a century, a computer should be able to pass a standard intelligence test (known as 'The Turing Test') in at least 30% of the tests performed. At the beginning of the twentieth century, few people shared Turing's confidence that a machine would pass the intelligence test in such a short period. However, currently many researchers are dedicated to the development of machine intelligence, which is the main objective of 'adaptive systems' and AI.

Bostrom (2014) has noted the difference between AI and human-level AI, and the fact the pioneers of AI didn't contemplate greater than human AI. Most importantly Bostrom (2014) notes how computers already outperform human intelligence in many domains, beating human champions in a wide range of games. The achievement of human-level AI, will eventually lead to a greater than human AI. Nilsson (2009) has noted Knuth's (1998) surprise regarding how AI has already succeed in many fields that require thinking, but how it has failed to do most of what humans do without thinking. Currently machines are far less intelligent than humans in general intelligence, even if they outperform humans in many other complex calculations. A key to feature to the development of artificial intelligence has always been to develop programs capable of learning, instead of pre-programmed machines, which was one of Turing's original ideas (Bostrom 2014).

Current artificial intelligence agglutinates a series of disciplines including machine learning. According to Domingos (2015) all knowledge comes from just three sources, genetics, experience and culture. Genetics is embedded in our bodies by evolution, like a computer operating system (OS). Experience is the learning since early childhood through our lives, the knowledge encoded in our neurons (computer user). Culture is the information embedded in each social structure that we belong, basically through other people, books, etc., which is specific to humans (software). Domingos (2015) has argued that recently there is a new source of knowledge, which are computers. In fact, computers can generate knowledge, through machine learning, by:

-Filling gaps on existing knowledge. Symbolists.

-Emulate the brain. Connectionist.

-Systematically reduce uncertainty, or the probability of what is true.

-Notice Similarities between the old and new.

-Simulate Evolution.

This thesis will use the last approach, the simulation of evolution to achieve machine learning. The term ‘adaptive systems’ covers many techniques that, applied to a problem, adjust the relative importance of the input parameters autonomously to get the resolution to the problem. Therefore, the objective is to achieve autonomous machine learning, which would lead to automatic programming.

In the 1950s Arthur Samuel raised the question of how can computers could learn to solve problems without being explicitly programmed it, which lead to the origin of evolutionary computation. Evolutionary computation it is based on replicating natural structures, by simulating evolution, with the aim of generating systems that will adapt to their environment in a similar way that nature does.

This thesis will focus on evolutionary computation techniques, which have so far been successful in problem solving, such as genetic algorithms (Holland 1975) and genetic programming (GP) (Koza 1989 & 1992), yielding excellent results in problem-solving for which traditional techniques didn't work.

Genetic programming (GP) has become a popular technique in recent years. GP is as an inductive algorithm by symbolic regression, that works by extracting the knowledge to solve a problem from a series of data obtained from a given problem. In order to understand GP it is necessary to understand how an evolutionary systems works. In particular, how the problems can be coded in the form of the individual's genetic material, the operators, and how the process of natural evolution works in the search for solutions.

2.3.2 Evolutionary Computation

Living organisms have exceptional skills for complex problem solving, far beyond the most complex computer programs and systems at the moment. To solve a problem through computation and prepare the algorithm can take months or years of intellectual effort, however organism change through mechanisms such as evolution and natural selection, in a process that takes billions of years and thousands of trials to achieve the right result. Evolutionary computation has its origin in the Darwinian evolutionism (Darwin 1859). The theory of the natural selection of Darwin argues that individuals in a population possessing the most advantageous characteristics will proportionately leave more descendants into the next generation, if those characteristics are due to genetic differences that can be transmitted to the offspring, over time there will be a tendency to change the genetic mix of the population, increasing the number of individuals with those characteristics. In that way, entire populations of living beings adapt themselves to the changing circumstances of its environment, resulting on living being's tendency to improve over time in relation to the surrounding circumstances.

Genetic algorithms allow the exploration of much more solutions than traditional computer programs. Nature has been practicing for millennia a combination of crossover and selection of lineages in search of better harvests, faster speeds, more intelligence, etc.

In most organisms, evolution occurs because of two primary processes: natural selection and sexual reproduction. The first determines which members of a population will survive to reproduce. The second guarantees that the mixing and recombination of their genes will be passed to their offspring. In the fusion of ovule and spermatozoa, homologous chromosomes stretch themselves and get attached to each other, in intermediate zones they interweave, thus exchanging their genetic material. Thanks to this mixture and genetic crossing, living beings evolve at a much faster than if each offspring just contained a mere copy of the genes of a single parent. Also, sometimes organisms are modified by simple mutation. Selection is a simple process: when an organism fails in some adaptation test, such as the recognition of a predator, the organism dies before it can reproduce, consequently, eliminating deficient or poor performance. Translating this logic to computer programs, for example, if a program has the function of ordering in ascending sequence a series of numbers, their operation will be verified, so that each number in the output list is greater than preceding, when this does not happen, the algorithm fails and is eliminated.

2.3.3 Genetic Algorithms

In the 1960's John H. Holland began to study the logical processes involved in adaptation, inspired by the studies of cellular automata by Burks (1960) and neural networks by Selfridge (1958). He noticed that simple rules could generate complex flexible behaviors, and visualized the possibility of study the behavioral evolution of complex systems (Holland 1975), initiating one of the most promising lines of study of AI, first published in the book 'Reproductive Genetic Plan', later popularized under the name 'genetic algorithm'.

Holland saw the process of adaptation in terms of a formalism in which programs or a population interact and improve based on a certain environment that will determine the appropriateness of its behavior (Holland 1992). Combining random variations with a selection process, should lead to a general adaptive system, (Holland 1995). Although conceived originally in the context of machine learning, the GAs have been extensively used in optimization (Quagliarella 1998), becoming an extremely popular technique in recent years. A GA is a search algorithm is based on the observations from sexual reproduction and the principle of the 'survival of the fittest', which allow species adapt to their environment and compete for resources. Applied to Computers, the algorithm works by the evolution of successive generations of individuals from the previous ones. Each generation consisting of a set of individuals (population) with a string of binary values each. Within this chain each individual is called a 'chromosome', each position within the chain is called 'gene' and the value within that position is called the 'allele'. A complete definition of a GA was proposed by Koza (1992): 'It is a highly parallel mathematical algorithm that transforms a set of individual mathematical objects with respect to time using operations modeled per the Darwinian principle of reproduction and survival of the fittest, after a series of genetic operations and more particularly sexual recombination. Each of these mathematical objects usually is a string of characters (letters or numbers) of fixed length that conforms to the model of the chains of chromosomes, and are associated with a certain mathematical function that reflects their aptitude'.

Operation of the GA.

Genetic algorithms establish an analogy between the set of solutions of a problem and the set of individuals of a natural population, codifying the information of each solution in a chain of values called chromosomes, consisting of bits or numbers (genes). The chromosomes evolve through iterations, called generations. In each generation, the chromosomes are evaluated using some measure of aptitude (evaluating the values that act as unknowns and assigning a value that determines the ability of that individual to solve the given problem). The following generations (new chromosomes), called offspring, are generated using two operators, crossover and mutation.

In the development of a new generation, reproduction is called the creation of new individuals from those already existing in the population that forms the generation. A very important factor in the operation of a GA is the selection of individuals for reproduction. For the GA to work, the Individuals (those whose fitness level is better) will reproduce more times than the others. Following this idea, multiple selection algorithms have been developed: Baker (1987), Booker (1982), Brindle (1981), De Jong (1975).

All existing selection algorithms are based on the same concept: to choose Individuals (either probabilistically or deterministically) by giving more possibilities to the best ones, but also allowing for the worst ones to be selected. If wasn't made that way, and only the best ones were chosen, the algorithm would converge prematurely making the whole population the same. The general functioning of the algorithm is shown in Fig. 35.

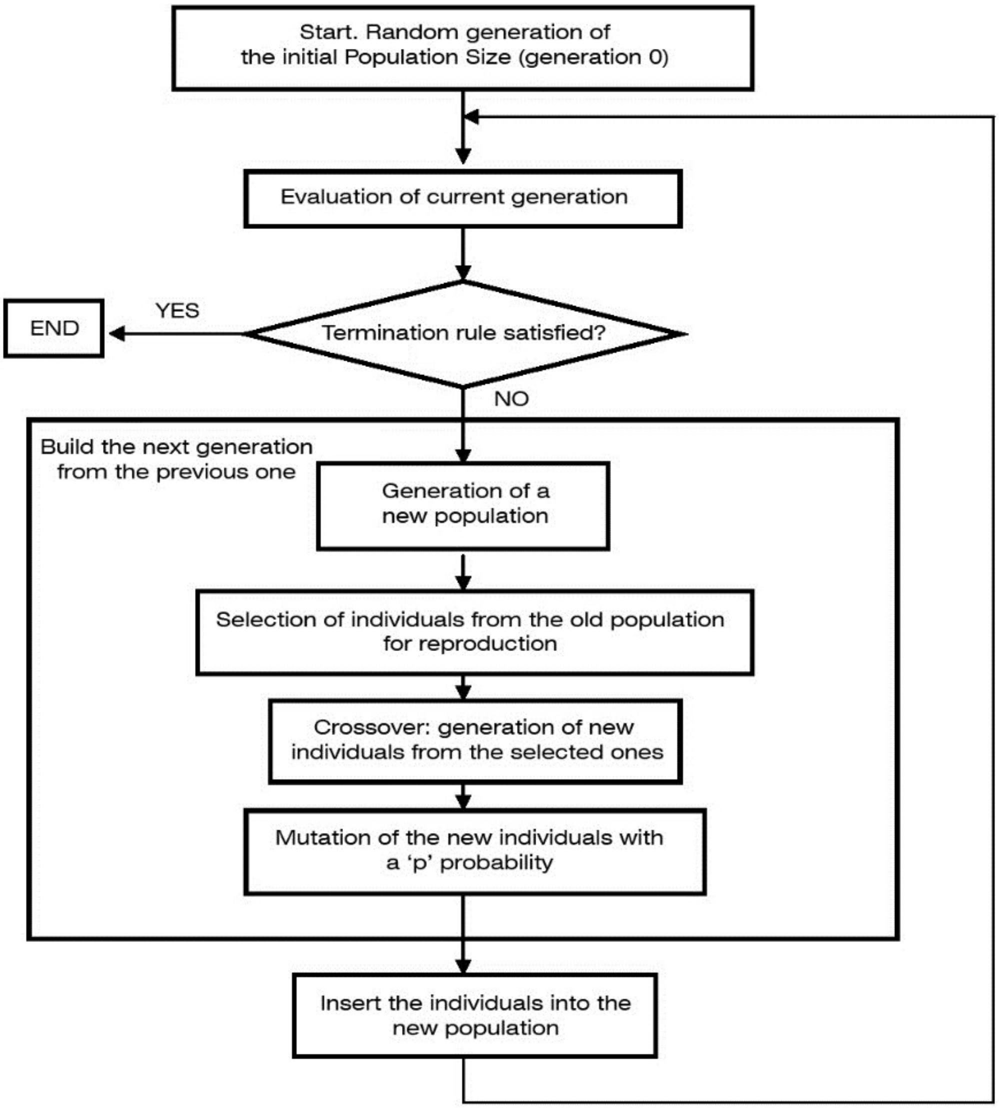


Fig. 35. General flowchart of a genetic algorithm.

After creating new individuals, and assigning a low probability (1% ~5%), each new individual undergoes a mutation process: the 'bytes' string is randomly changed. At the end, new individuals are inserted into the new generation creating a new population that will form the next generation.

This process is repeated until some pre-set stop criteria is satisfied, Such as:

- There is an individual in the population who has reached good enough results.
- The maximum number of pre-set generations has been reached.

- The population has converged and all individuals (or a high percentage) are The same.

Since each individual represents a possible solution to the problem, the existence of a large number of individuals in the population implies that the algorithm performs a search in many different regions of the space at the same time.

An additional advantage of GAs is that troubleshooting uses a fit measure to direct the search, and requires no specific knowledge of the search space, being efficient in spaces that have jumps, noise, valleys, etc. As each individual within the population conducts the search, the GA performs a parallel search in numerous points of the space of states with numerous directions of search, Fuchs (1998).

2.3.4 Genetic Programming

Genetic programming (GP) emerged as an evolution of the traditional GAs, by maintaining the same principle of natural selection. What GP does is to provide solutions to problems by creating programs and algorithms to solve them.

GAs provide results by the adjusting a series of numerical values encoded as a ‘chromosome’ that solves the problem. In the case of GP, the results, can be a program, another algorithm, or even a mathematical expression that solves the problem, depending on the technique used. The fundamental difference Between a GAs and GP is the way solutions are coded. While GAs work through a sequence of values, GP is represented by a ‘tree’ structure, coded in the form of a mathematical formula. The nodes of this ‘tree’ like structure are the operators (arithmetic operations, trigonometric functions, etc.) and the terminals or ‘sheets’ are the constants and variables.

The first attempts to evolve programs date back to the 1950’s and 1960’s (Friedberg et al 1959) (Fogel 1964), but it was not until the 1980’s that satisfactory results were achieved. Hicklin (1986) and Fujiki (1986) used expressions in LISP (List Processing) to represent programs whose purpose was to solve problems of game theory. Cramer (1985) and later Koza (1989) independently proposed the use of tree representation, where the crossover operator was implemented exchanging sub-trees between different programs of a randomly generated population. Koza's (1992) book: 'Genetic Programming. On the programming, of computers by means of Natural Selection', laid the basis for genetic programming. The techniques he developed have been used in a wide range of applications, such as image compression, design of electronic circuits, pattern recognition, robot movements, etc. GP was further improved through the incorporation of Automatically Defined Functions (ADF), which can be used as sub-routines and notably increment the capabilities of GP to automatically generate programs.

Among the reasons why GP has become a powerful and easy problem solving tool are:

- Implements a stochastic search process, performing searches in several regions of the space, with explorations in new sectors. It has proven itself as an efficient search algorithm, suitable for problems in which the search space has many local minimums, valleys, etc., for problems that algorithms, such as gradient minimization, would not reach a solution or would have a hard time reaching the solution.
- Unlike Traditional GAs, the coding of each solution (usually called the chromosome) is done in terms of the problem itself, determining a series of unknowns to which the algorithm will attempt to assign optimal values to solve the problem. In GP the solution sought will be an algorithm and a tree, then transformed into a mathematical expression. For example, from a series of empirical values of a phenomenon, the algorithm will try to find the explanation of the phenomenon in terms of a mathematical expression.

- Working with these types of structures minimizes and facilitates the necessary analysis for the codification. Since it is going be done in a similar way that the problem to be addressed, being very versatile and fit to solve a variety of problems.
-They are inherently parallel algorithms, since they are based on realizing calculations in a massive and independent way (by evaluating many different and independent solutions).
-Solutions can be reached in a hierarchical way, priority can be given to some factors in relation to others when carrying out the process of search for the expression, as well as determining the complexity of the solutions obtained.

2.3.4.1 Operation of GP

The operation of GP is similar to the operation of a GA, which is based on successive iteration of generations from the previous ones (Fig. 36) Koza (1992).

After the initial creation of trees, which will generally be random, successive generations will by copies, crossovers and mutations of each one of the individuals from the previous generation.

The first step in the operation of the algorithm is the generation of the initial population. To create generation 0, each tree will be generated randomly, depending on the algorithm, considering the restrictions that exist in the trees. Since the trees are random, the individuals in this first population, tend to represent bad solutions to the problem.

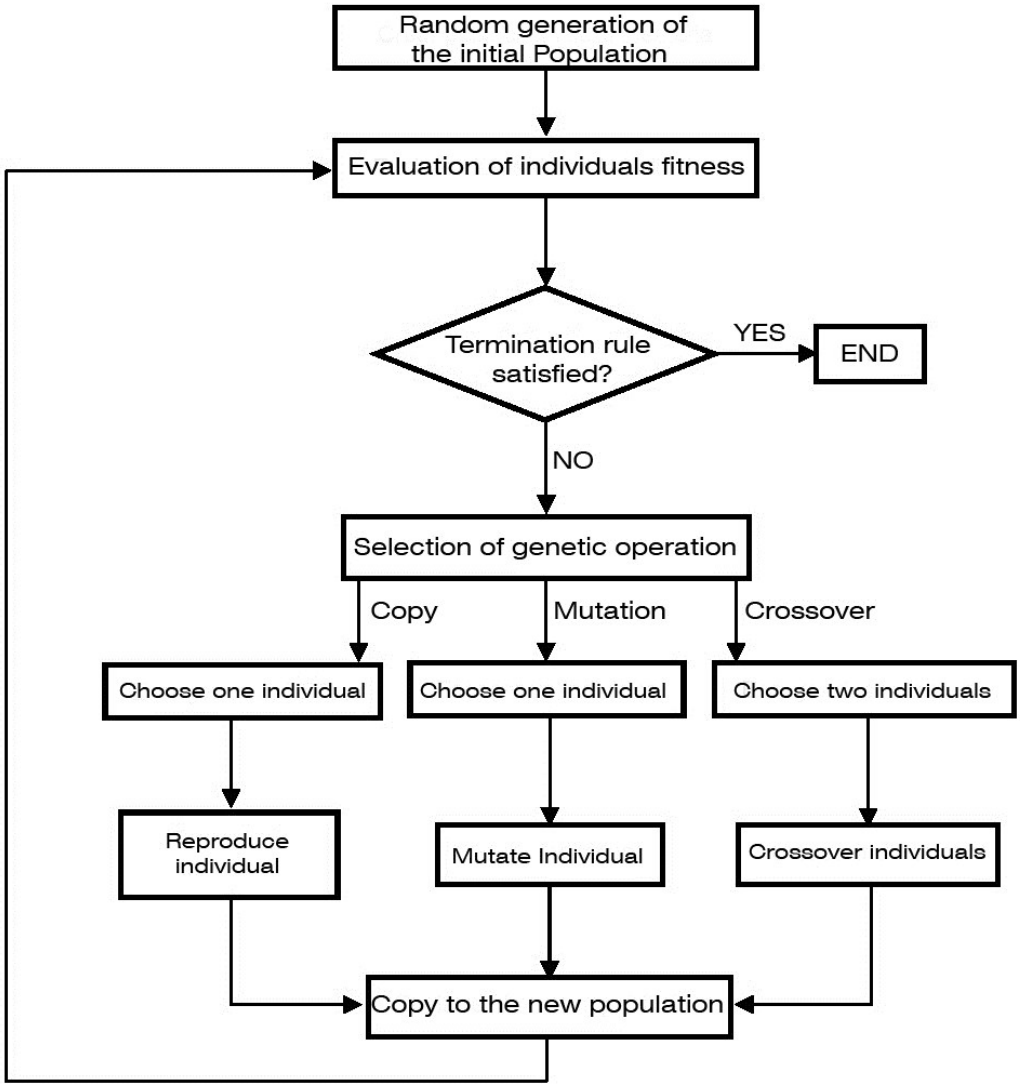


Fig. 36. Flowchart of genetic programming.

For the creation of a tree there are a great variety of algorithms, but the most used are (Koza 1992): the partial, complete and intermediate algorithms.

-The ‘partial’ creation algorithm generates trees whose maximum height does not exceed the one specified.

-The ‘complete’ creation algorithm generates trees whose leaves are all at certain level, since generates complete trees.

-The ‘intermediate’ creation algorithm is a mixture of the two previous ones, to achieve a greater variety and genetic diversity in the initial population. This algorithm is based on executing the two previous ones by alternating them and making different heights to create all the elements of the population.
All these algorithms are random based, and the only intervention of the user is the initial introduction of the input elements.

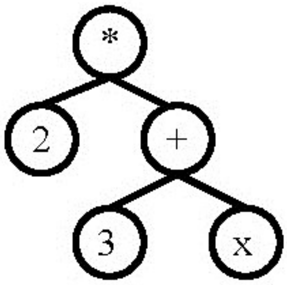


*Fig. 37. Tree for expression 2 * (3 + x).*

The tree-based representation has two types of nodes:
-Terminals, or leaves of the tree, are the ones without children. They are usually associated with constant or variable values.
-Non-terminals, are those with one or more children. Generally, associated with operators of the algorithm to be developed. On Fig. 37 an example of a tree is shown, representing the expression F (x) = 2 * (3 + x). It has as two non-terminal nodes corresponding to the product and the sum, and three terminal nodes corresponding with the values 2 and 3 and the variable x.

A fundamental part of the operation of the GP is the specification of the sets of terminal and non-terminal elements before the beginning of the evolutionary process. By specifying nodes, the algorithm will build the trees. Therefore, it is necessary a minimum process of analysis of the problem to configure the algorithm, since the operators or functions that can be used need to be specified (sine, cosine, exponential, etc.). Generally, it is advisable to adjust only the necessary number of operators, since the addition of non-necessary items won't result into a not found solution, but it will result into a longer processing time.

When specifying the sets of terminal and non-terminal elements, they need to meet two requirements: sufficiency and completeness (Montana 1995). The sufficiency requirement states that the solution to the problem must be specified within the set of operators. The requirement of completeness states that it should be possible to build correct trees within the specified operators. Since tree-building is a random process, many of the trees built will not be correct, not for not following the rules, but by the application of the operators (non-terminal nodes) to elements that they are not in the domain. A good example of this, is the division operator (equation 4), whose domain is the set of real numbers except the value zero. Expanding its domain, a new operator is defined (%):

$$\%(a,b)\begin{cases}1 \; if \; b=0\\ \frac{a}{b} \; if \; b\neq 0\end{cases}$$

Equation 04. Operator %

This new operation is called a protected division operation. In general, when a new operation is created, extending the domain, it's called protected operation.

There are two main types of restrictions:

-Tipped.

-Maximum tree height.

To establish syntactic rules in tree creation, it is possible to specify typing rules (Montana 1995). The type of each terminal node is established, and for non-terminals the type that each child should have. By doing so the structure that the trees need to follow is specified. By specifying the type of node, the algorithm building grammar to create tress is defined, allowing the trees to have the desired structure.

The most used type of operators is related to the realization to arithmetic operations: real and integers. However, other ones are also used such as Boolean, or the sentence type. This last one is used to develop a program as a sequence of commands, that designates nodes with no return, basically their evaluation is based on the effects made by their execution.

On the other hand, height restrictions avoid the creation of trees too large and forcing the search for solutions within a prearranged size. This restriction prevents trees from having a lot of redundant code, as well as excessive tree growth (Soule and Foster 1997), (Soule 1998).

2.3.4.2 Genetic Operators

To make new genetic operators modifications are applied to the trees. These operators are the result of adapting existing GA to GP, by modifying the GAs by adapting them to a tree structure encoding. The most used operators are:

-Crossover.

-Reproduction (copy).

-Selection.

- Mutation.

- **Crossover,** is the main operator. Two individuals from the previous population are randomly combined to generate two new individuals. After selecting two individuals as parents, one random node is selected as the first and another one as the second. For the exchange, not to break any of the restrictions: the nodes must be of the same type and the new trees must continue to maintain maximum height. The crossing between the two parents is affected by the interchange of the subtrees selected in both parents (Luke and Spector 1998). There are adaptive variants of this operator (Angeline 1996) on which the own algorithm is modified, as well as numerous variants of the algorithm, (Aguirre et al 1999), (Pereira et al 1999).

- Reproduction (copy), is the copy of individuals to the new generation. This operation is asexual in the sense that an individual is generated from a previous individual. Reproduction and crossover, are the most widely used operators. Crossover is more frequent, the percentage of new individuals generated from crossovers is usually greater than 90%, while the rest are generated by copies. When the number of individuals generated by copies increases, also the danger of an individual's predominance over the rest of the population increases, and finally after several generations the whole population can converge towards that individual. This is an undesirable situation, since has completely lost the genetic diversity that it had at the beginning and this entails that the search in the space only will take place in a certain zone, which is the opposite of what a GA intends. The crossover operator, is therefore the main operator used for the generation of new trees (Poli and Langdon 1998).

- Selection of individuals, happens to avoid an individual's predominance in the reproduction process by crossing or copying. It is necessary to regulate the selection of individuals to whom these operators apply. There are many selection algorithms, to perform the task of choosing which individuals will reproduce and which ones will not. In general, all the selection algorithms are based on the same idea: the most fit individuals will be much more likely to be chosen for reproduction, but without completely eliminating the possibilities of the less capable individuals to be chosen, otherwise the population would converge in just a few generations. The measurement of the goodness of an individual, this is assessed by a level of adjustment or aptitude. Based on that level it will be decided which individuals are chosen. Some of the most used algorithms for such a purpose are the selection by tournament and selection by roulette. In tournament selection (Wetzel 1983), a random number of individuals are chosen among the whole population (typically two individuals), and the best one is selected. This selection method is widely used, and can regulate the selection pressure exerted on the population by varying the number of individuals participating in the tournament. In this way, if a low number of individuals is involved, little pressure is exerted and the least fit have more opportunities to be selected. As the number of individuals grows, the best ones will be selected more frequently. In addition, it is possible that an individual gets selected several times.

In roulette selection (De Jong 1975), all individuals in the population are arranged in a roulette each one occupying a part proportional to the adjustment level of the individual compared to the adjustment level of the entire population (sum of adjustments); basically, the best individuals will take up more space in the roulette. To perform the selection, the roulette is simply rotated and the selected individual is returned. This method is widely used for its simplicity and good results. However, it presents the problem that when selecting the same individual several times, the best individual is chosen many times and ends up predominating in the population.

There are many other algorithms, on which the number of times an individual is selected is chosen deterministically, in that way avoiding the predominance of an individual. Each of these algorithms presents variations regarding the number of times that the best and the worst individuals will be chosen. A pressure on the search spaces will be placed in the area where the best individual is allocated, or it will be a tendency to divide the search in the space areas, but without stop trying to look in the best area. Some of these algorithms are: surplus stochastic (Brindle 1981), (Booker 1982), universal stochastic (Baker 1987) or sampling deterministic (De Jong 1975).

- Mutation operator causes the variation of a tree in the population. This operator is usually used with very low probability (less than 10%), and it is used before introducing an individual in the new generation. There are two main types of mutation: mutation in which just a single node changes and a mutation in which an entire branch of the tree changes.

  The first case is known as a point mutation; the mutation happens in the following way:

  1. A tree node is randomly chosen.

  2. Randomly a node is selected from the set of terminals or non-terminals, from the same type as the one previously selected, with the same number and type of children.

  3. The old node of the tree is exchanged for the new node, keeping the Same children as the old.

  Since each branch of the tree represents a solution to a sub-problem and the non-terminal that joins them represents the way to combine those solutions, if this type of mutation is done on a non-terminal element will be causing the solutions to be combined in different ways. This type of mutation rarely used.

The mutation operator causes an individual to jump in the states of the space, starting a different search in another zone. Most mutations are destructive, meaning that the individual gets worse, therefore they are used with a very low probability, to get genetic variety. There are some studies on evolution without use of crossover, in which mutation plays a fundamental role, (Chellapilla 1997), by using two mutation types, but the results are not as satisfactory as using crossover.

- **Evaluation** is the quantification of the goodness of a given individual. The evaluation value represents how well the tree have solved the problem. Adjustment is probably the main concept of Darwinian evolution within GAs and GPs, and refers to an individual's ability to compete in an environment for the available resources. Goldberg (1989) described the adjustment function as a measurement of the goodness to be maximized.

  In a genetic algorithm, the competition is based on the performance of the chromosome within the domain of the problem. An appropriate scale is determined for a task such as 'time before failure' (Randall et al 1994), or 'time before stabilization' (Koza 1990). After having applied a chromosome to the problem, a value that reflects its performance is assigned to it. By doing this, when the entire population has been tested, the relative ability of each chromosome can be quantified. There are three different types of assessment of the adjustment measure (Koza 1992):

  -Standardized. This type of adjustment [s(i,t)] measures the goodness of an individual in the generation 't', values close to zero indicate a good value of adjustment and distant values a bad individual. Therefore, in a generation ‘t’ an individual ‘i’ will be worse than another ‘j’ if [s(i,t)>s(j,t)]. This measure is very useful in problems in which the quantification of the adjustment level of the individuals is based on penalties, such as the error in induction of formulas, mean square error, number of executions required to find the solution, etc.

-Tight. This value is obtained as follows:

$$\mathrm{a}(i,t) = \frac{1}{1 + s(i,t)}$$

Equation 05. Adjustment formula.

With this measurement, the goodness is quantified between 0 and 1. The value 1 corresponds to the best individual and 0 to the worse one.

-Normalized. It is a comparative adjustment value of the individual within all the population. It is obtained from the ‘*normalized fit formula’,* given a M population size:

$$\mathrm{n}(i,t) = \frac{a(i,t)}{\sum_{k=1}^{M} a(k,t)}$$

Equation 06. Normalized adjustment formula.

This value will be between 0 and 1. This type of adjustment indicates the level of goodness within the population. In this case, the objectivity of evaluation is gone, and a value close to 1 no longer indicates that this individual represents a good solution to the problem, just that this individual represents a better solution compared to the rest of the population. This value is used for selections proportional to the adjustment, such as roulette. In this case, the ratio of the roulette occupied by an individual will be this value, since the sum of all the values will be 1.

2.3.5 Interactive Evolutionary Computation: NEvAr

Interactive evolutionary computation (IEC), is a term that denotes the intervention of a human in an evolutionary computer process, typically by manually setting the fitness values. Decisions are based on an individual's own preferences, who will guide the evolution of a system.

Instead of assigning a fitness function into the computer programming a human will act as the fitness function. The idea is combining the GA and GP and assigning the fitness functions to a person who will decide the fitness values. The disadvantages of this methodology are several such as the human fatigue, subjectivity, low speed and manpower cost compared to computers. IEC however are efficient for small populations, that need few interactions. Also for those cases where the fitness functions are hard to stablish, such an artistic or an individually customized evolutionary process.

Several IEC methodologies have been proposed, for both single and multiple evaluators. Multi-user interactive IEC systems can be used as a collaborative tool, were many users can define the fitness functions in a collaborative way, this approach to evolutionary computation is based on the work of Sims (1991).

Among some of the IEC methods are, interactive evolution strategy, interactive genetic algorithms, interactive genetic programming and human-based genetic algorithms. This thesis is particularly concerned about GAs, GPs and interactive genetic algorithms (IGA). IGAs are basically GAs or GPs that use human evaluation.

IGAs are in very useful for subjective evaluation processes, based on an individual preference, such as evolutionary processes involving art, music, designs, colors, etc.

'Neuro Evolutionary Art' (NEvAr) will be used later in this thesis for a case study. NEvAr is a research project of the artificial intelligence Laboratory of the University of Coimbra, developed by Machado and Cardoso (2000), which purpose is autonomous image and music generation, by using evolutionary computation. NEvAr is inspired in the work of Sims (1991) and Dawkins (1987), and allows the evolution of populations of images based on the user's preferences, by implementing a parallel evolutionary algorithm (Fig. 38).

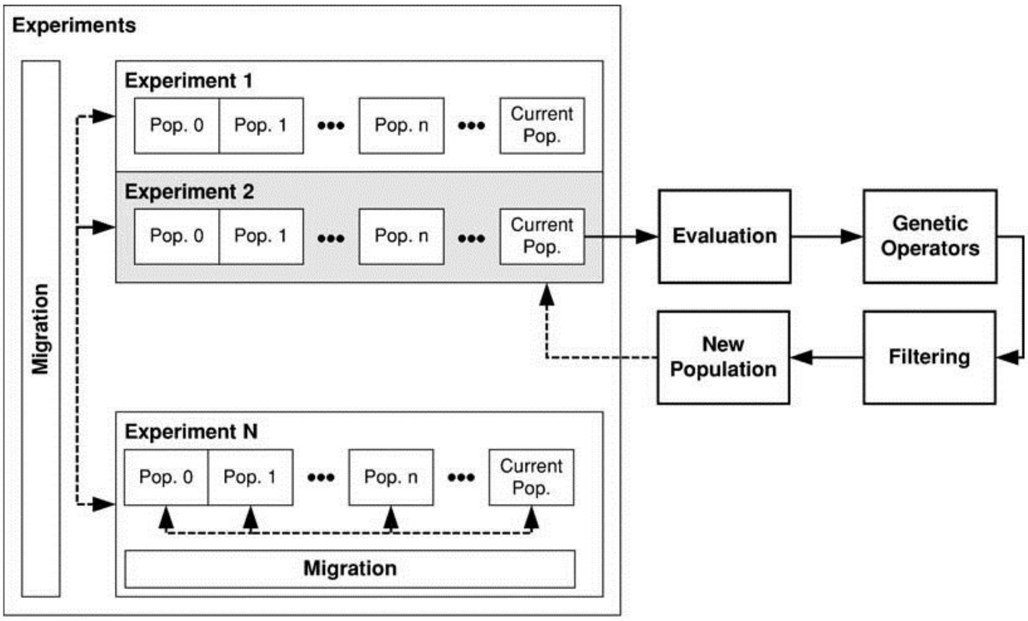


Fig. 38. NEvAr GP evolutionary model Flowchart. Source: Machado and Cardoso (2002).

In NEvAr the individuals are images, which is the phenotype. The genotype are trees made of simple 'functions' (such as arithmetic, trigonometric and logic operations) and 'terminals' which are composed of a set of variables '*x*' and '*y*' and random constants (Fig. 39).

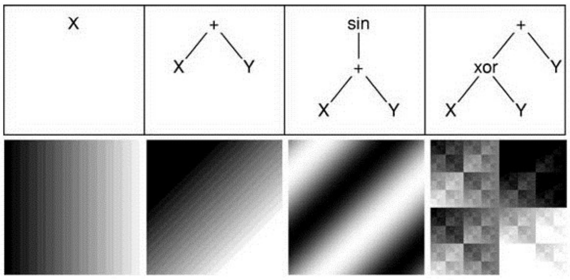


Fig. 39. NEvAr, examples of some simple functions and the corresponding images. Source: Machado and Cardoso (2002).

NEvAr generates images through mathematical expressions, by genetic operations of recombination (Standard GP crossover operator (Koza 1992)) and mutation performed at the genotype level. The interpretation of the genotype will result into the phenotype which are the individuals (Machado and Cardoso 2002).
Using NEvAr, artists can generate new images, but the artist is not responsible of the generation of the images themselves, it will just be the evaluator of the evolutionary process (Machado and Cardoso 2000). NEvAr is a user based interface, that requires the intervention of a human, and will develop images based on the user's preferences. As other evolutionary tools NEvAr requires a learning period, and the first results will be typically not satisfactory.

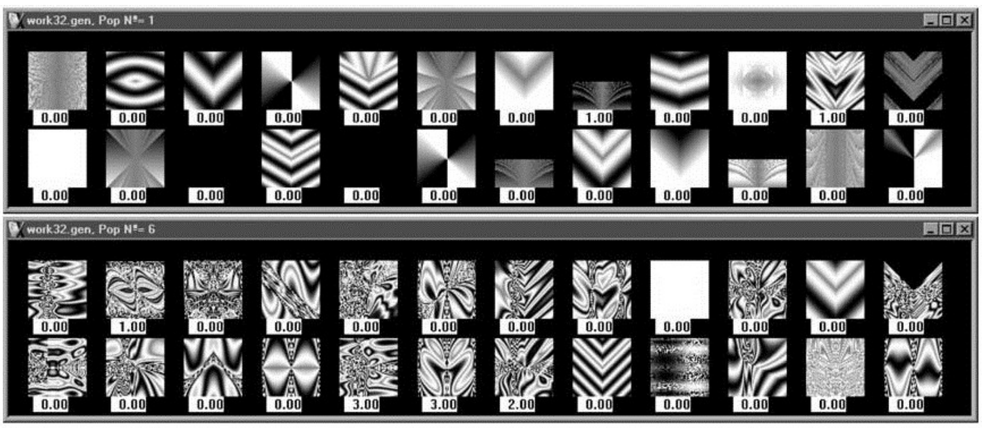


Fig. 40. NEvAr interface, fitness values assigned to the images by a user on populations,1 and 6. Source: Machado and Cardoso (2002).

The process consists of several steps, which are: discovery, exploration, selection and refinement. The idea is that the user stars with a concept or idea, then a random population is generated, to then recombine certain selected individuals again and again. This process happens by the user assigning a fitness value to the preferred images (individuals), as shown in Fig. 40.

By successive recombination of certain images the results will eventually lead to a final set of images. During the process the settings can be modified for each stage, as well as the number of individuals on a population, increasing or decreasing its numbers. Once some satisfactory results have been achieved, a specific image can be used to refine the process, by generating again random variations, and recombining the most satisfactory individuals by assigning a fitness value. All this process requires some skills and aesthetic preferences from a specific user, leading eventually to some satisfactory results based on the preferences of that user. Fig. 41 shows a final set of images resulting of multiple recombination through discovery, exploration, selection and refinement, resulting into a final population of successful images for a specific user (Machado and Cardoso 2002).

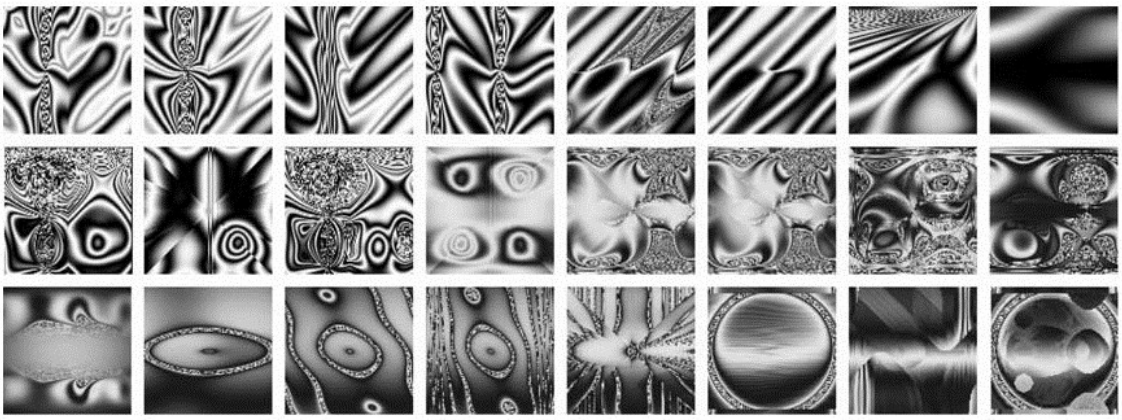


Fig. 41. Example of best individuals from several independent runs. Source: Machado and Cardoso (2002).

3 HYPOTHESIS

3.1 Main Hypothesis.

On chapter 2 the basic notions of CG, point clouds, polygon meshes, NURBS, PBR, evolutionary computation, GA, self-organization and morphogenesis were introduced. All the previous processes, formats and techniques have been recently developed. The objective of the thesis is to use some of the latest computer technologies and processes to propose methodological strategies to reformulate the basis of architectural and urban design representation, and ultimately the generation, by using artificial intelligence and in particular evolutionary computation techniques.

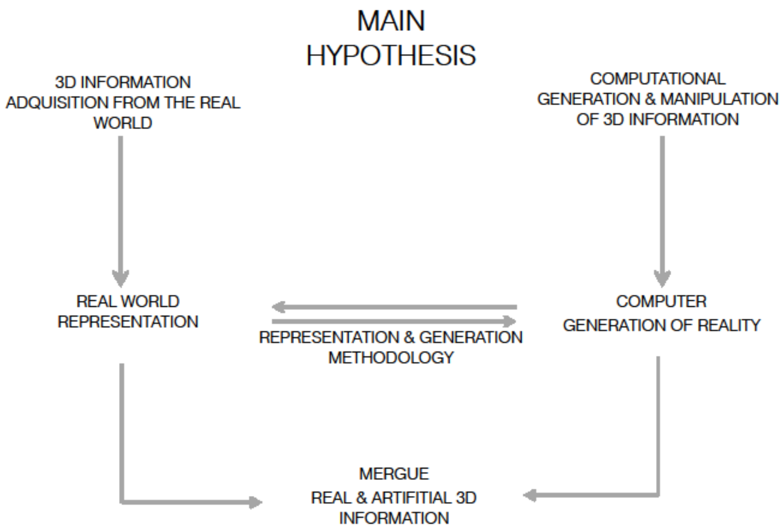


Fig. 42. Flowchart showing the main hypothesis concepts.

This doctoral thesis main hypothesis is that by introducing recently developed technologies and advanced computation techniques with a promising future, a greater level of integration between 3D models obtained directly from the 'real world' and 'artificially generated' 3D models that do not exist in the 'real world' can be achieved. The proposed methods and techniques will contribute to further blur the boundaries between 'real' and 'artificially' generated architectural objects and urban environments. The main concepts underlying the thesis hypothesis are shown on Fig. 42.

Recent LIDAR technologies can obtain 3D data from the 'real world', in an accurate, precise and realistic way, in a 3D model output. Recent computation techniques can generate precise 'artificially generated' 3D models. By using artificial intelligence 3D models could mimic more accurately real existing conditions, such as irregularity, growth or diversity. By combining these processes of obtaining real data and artificially generating data, more realistic 3D models could be obtained, not only from a representational point of view, but also from generative and morphological point of view.

The main hypothesis postulates that mixing and blurring the visual and geometrical boundaries between real objects and 'artificially generated' geometries could lead to methodological changes in architecture and urban design. The research process will be focused in three secondary lines of research, all of them based on the use of new technologies. The following techniques will be further studied, developed and tested through case studies:

3.1.1 Sub-Hypothesis 1.

Digital representation of real objects from precisely acquired point cloud LIDAR scans from the 'real world', might be a more efficient and accurate way to model and represent 3D architectural and urban objects from the 'real world', than other commonly used formats such as polygon meshes and NURBS. The use of Point-based Rendering (PBR) could help to better combine 3D objects resulting of LIDAR scans (real 3D objects) and 3D objects manually modeled (artificial 3D models). This methodological process could lead to time saving, computation resources savings and most importantly to a better resolution and greater accuracy of 3D renderings, when combining real and 'artificially generated' environments.

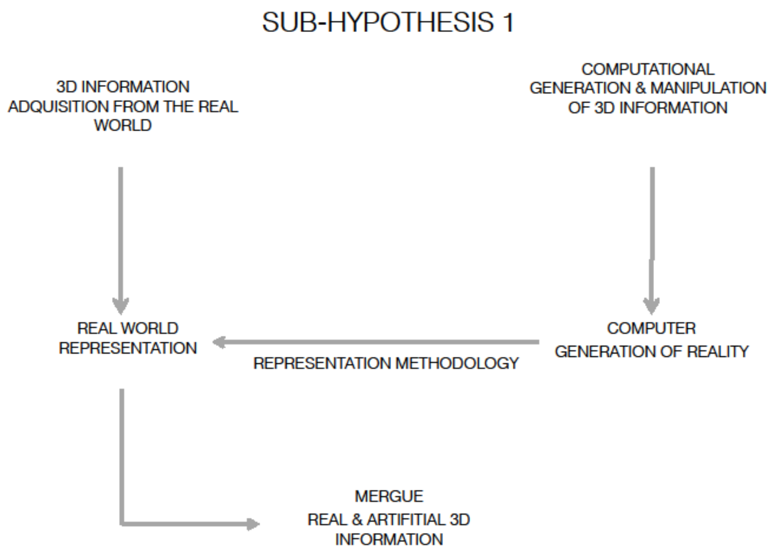


Fig. 43. Flowchart of sub-hypothesis 1.

3.1.2 Sub-Hypothesis 2.

The use of artificial intelligence techniques could assists in the modelling of artificial objects to be more realistically, by generating diversity and complexity (imperfections), as found in natural objects. The use of artificial intelligence techniques could also assist in the design process of generation of new 3D artificial objects and designs, using computational design techniques that given a problem will be able to auto-generate design solutions.

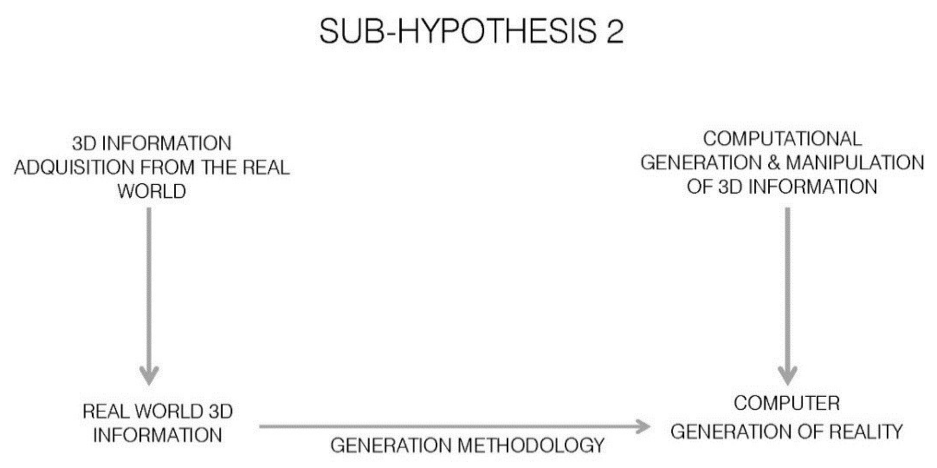


Fig. 44 Flowchart of sub-hypothesis 2.

3.1.3 Sub-Hypothesis 3.

The use of artificial intelligence techniques could assist to simulate the growth and evolution of real objects, such as self-organized systems, further merging the boundaries between real and 'artificially generated' objects. Human made complex self-organized systems, such as cities are in constant evolution. Biological processes such as morphogenesis could be successfully used to auto-generate, evolve or predict the evolution of those systems, blurring the boundaries between real growth and simulated growth.

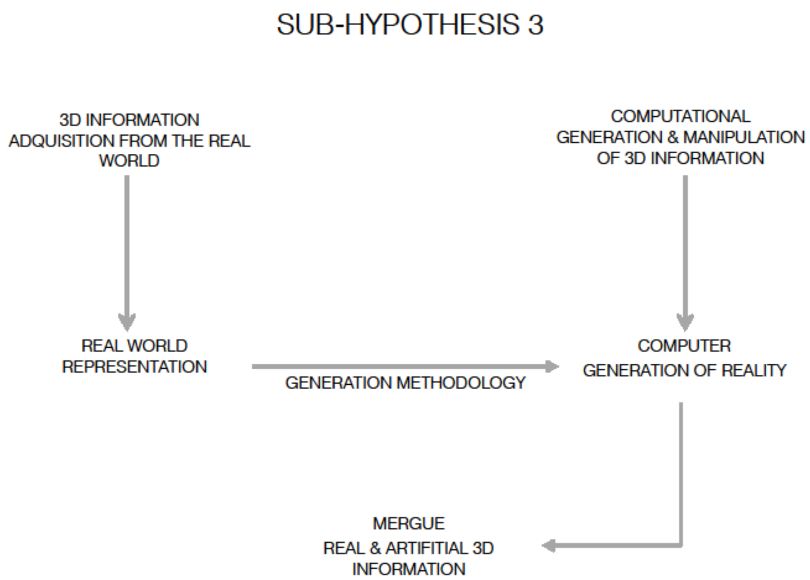


Fig. 45. Flowchart of sub-hypothesis 3.

The previously mentioned techniques will be applied and tested into several 3D models of different complexity, then the results will be evaluated, to draw the final conclusions.

4 METHODOLOGY

4.1 Divide and Conquer Methodology.

Novel technologies such as LIDAR scanners and relatively new computation techniques such as point cloud representation, PBR, morphogenesis and evolutionary computation applied to architecture and urban design will be tested and evaluated. The research will be done through experimental case studies that will test the main hypothesis, by proposing specific methodologies and algorithms. The research methodology used will be Divide and Conquer (D&C), which is based on multi-branched recursion, by dividing a complex problem into related sub-problems or parts, until the problem is simple enough to be solved directly. Once individual solutions are found, they can be combined to solve the initial complex problem, into a single solution. D&C is often used for algorithms to solve all kind of problems. The first description of the algorithm was made by Mauchly (1946), who in the same year co-created with Eckert (1946) the first general purpose electronic digital computer. According to Knuth (1998) the idea of using a shorter list of items to solve a problem dates back as far as to 200 B.C. Babylonia, and has been used for centuries in a non-algorithm form. Stake (2005) refers as a 'collective case study' a process of elaborating multiple projects because 'understanding them will lead to better understanding, and perhaps better theorizing, about a still larger collection of cases'.

The main hypothesis will be tested by dividing it into the three sub-hypotheses, that will focus on a specific sub-topic. The secondary sub-hypotheses will explore the subjects of representation, morphogenesis and evolutionary growth, of 'real' and 'artificially generated' 3D models. First the hypothesis will be tested on small and less complex objects, to increasable apply the hypothesis to larger and more complex 3D objects.

The ultimate purpose is to validate the main hypothesis, which is the use of new technologies and techniques (LIDAR, PBR, point clouds and artificial intelligence) to further blur the boundaries between 'real' and 'artificial' 3D objects, as shown on Fig. 46.

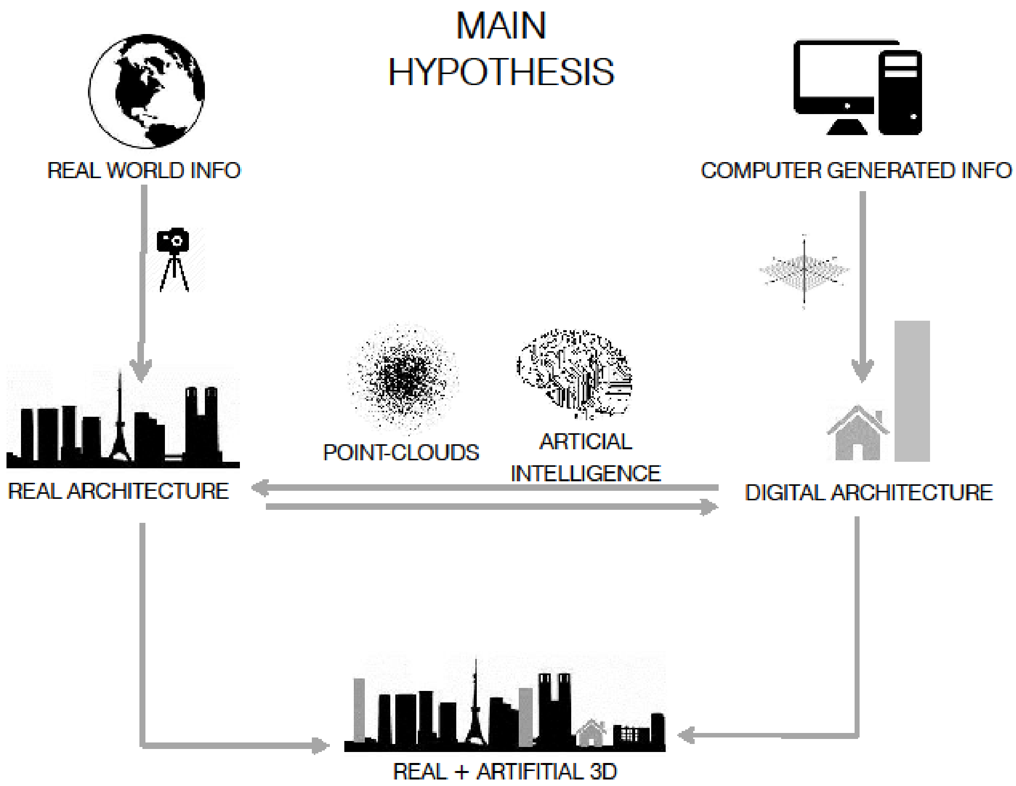


Fig. 46. Diagram of the main hypothesis concepts.

4.1.1 Merging Real and Artificial.

The first step will be to use data from 3D objects from the 'real world', obtained through LIDAR scans, in point cloud format, to then merge them with 'artificially generated' 3D objects. For such a purpose a conversion algorithm will be proposed, to later combine the 3D models using PBR techniques.

The methodological process will follow the next steps (Fig.47):

- 'Real world' Environments will be scanned by using LIDAR technology.
- A novel PBR technique will be tested to render the scanned objects.
- A manually modelled 3D object will be created.

-An algorithm to convert NURBS and polygon meshes into point clouds will be proposed and tested.

-'Real world' and 'artificially generated' objects will be combined, and then rendered using the novel PBR technique.

- The results obtained will be evaluated.

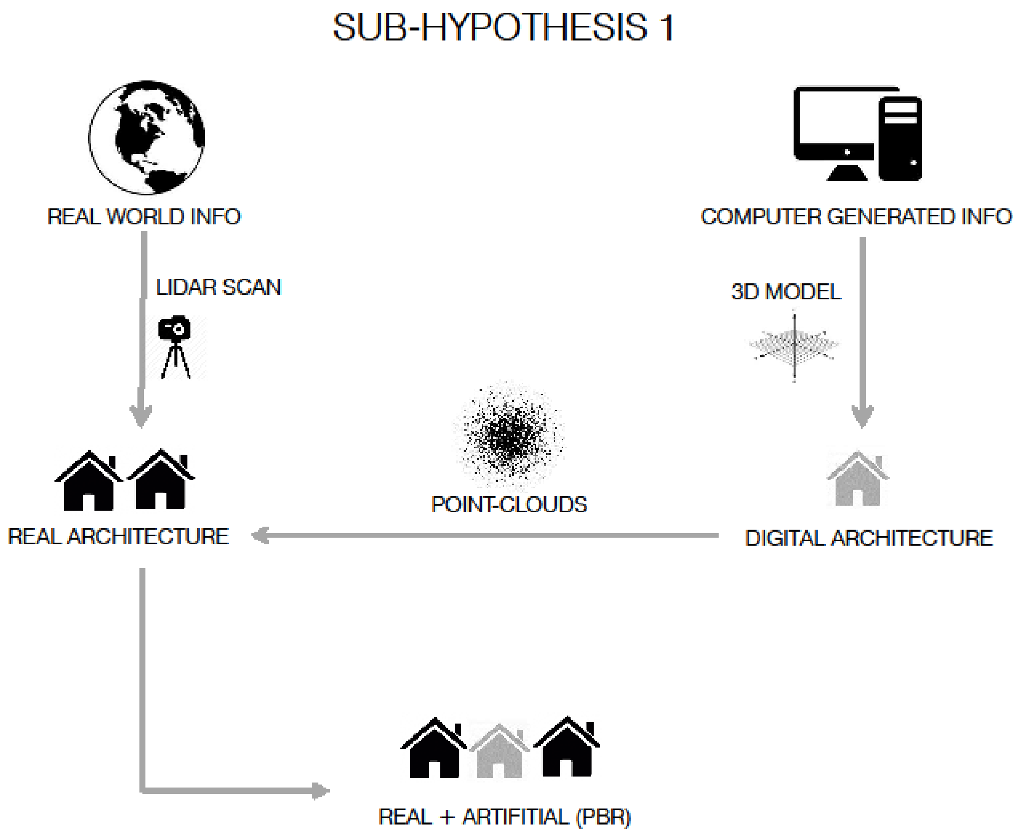


Fig. 47. Diagram of sub-hypothesis 1 methodology.

The case studies will compare the use of point-based geometry instead of the most commonly used formats of polygon mesh and NURBS, and whether is it a more efficient approach to combine 'real' and 'artificially' generated 3D models.

As a part of the case study an algorithm to translate designs generated as NURBS into point clouds will be proposed. This algorithm will enable a smooth data flow between meshes, NURBS and LIDAR scans.

4.1.2 Auto Generation of Diversity through Morphogenesis.

A second set of cases studies will explore the sub-hypothesis 2., creating diversity and new designs from 'artificially generated' 3D objects, using evolutionary morphogenesis, in that way achieving more realism.

The introduction of artificial intelligence techniques and genetic morphogenesis algorithms will be tested for the automatic generation of complex 3D geometries to create more realistic 3D objects. The case study will explore the application of evolutionary algorithms into the further manipulation of point-based geometry. Typically, geometrical diversity and imperfections in manually generated 3D models is achieved by manually modelling each sample. The objective is to test new ways to auto generate diversity of samples from an original 3D object, through evolutionary computation. Evolutionary computation will also be tested for the generation of new designs, from on pre-stablished parameters.

The following methodological process will be used (Fig. 48):

- An object from the 'real world' will be modeled in 3D.

-The object will be transformed into a point cloud.

- An evolutionary morphogenesis algorithm will be proposed and applied to its surface to achieve more geometrical realism (imperfections).

-The algorithm will create new populations (samples), based on the user's selections to be genetically recombined into the generation of new samples.

- The evolutive process will be repeated until the resulting geometrical outcome resembles the real original sample.

- The morphogenesis process will be then tested for the generation of new designs.

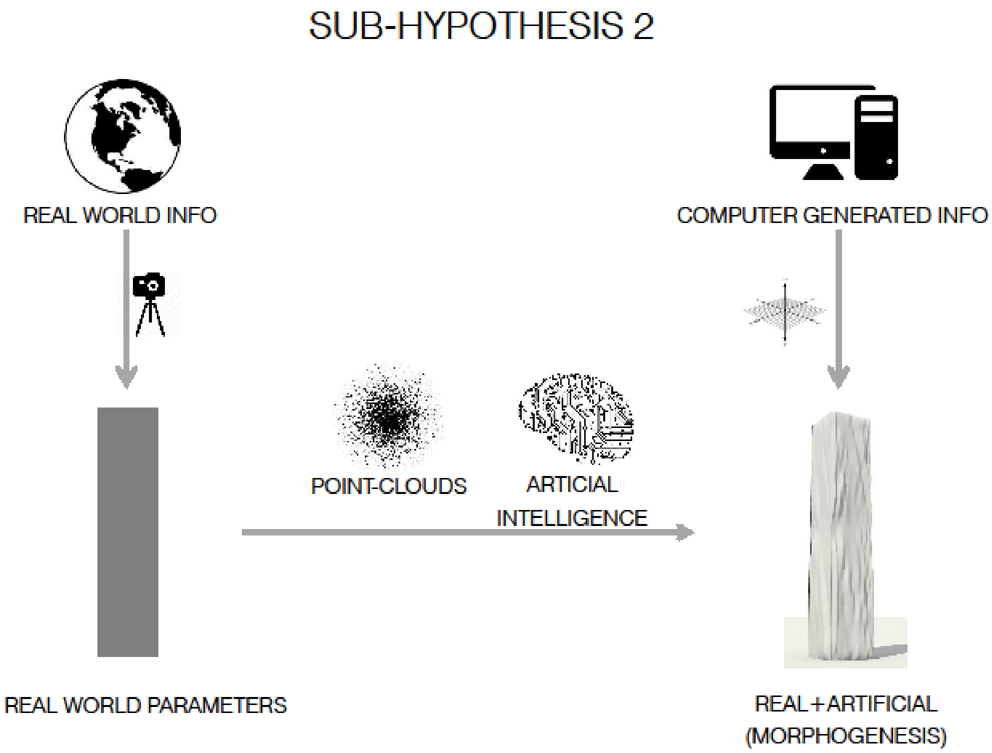


Fig. 48. Diagram of sub-hypothesis 2 methodology.

4.1.3 Evolutionary Growth of Self-Organized Systems.

The case study will test the use of evolutionary algorithms to predict the evolution of a real man-made self-organized system: A City. The idea is to predict future scenarios of urban vertical growth. The objective of the case study is to develop a computer model that can simultaneously determine the most likely location, height and number of new skyscrapers. Regarding inputs, it will make use of various criteria related to urban regulations, a survey of current buildings and economic indicators to the case study area to simulate future growth patterns. The proposed model is divided in two parts: the first is a parametric process, which will determine a probabilistic map for the allocation of new buildings; secondly, there will be a genetic algorithm based on economic data that will determine the estimated number of buildings built per year as well as their average height.

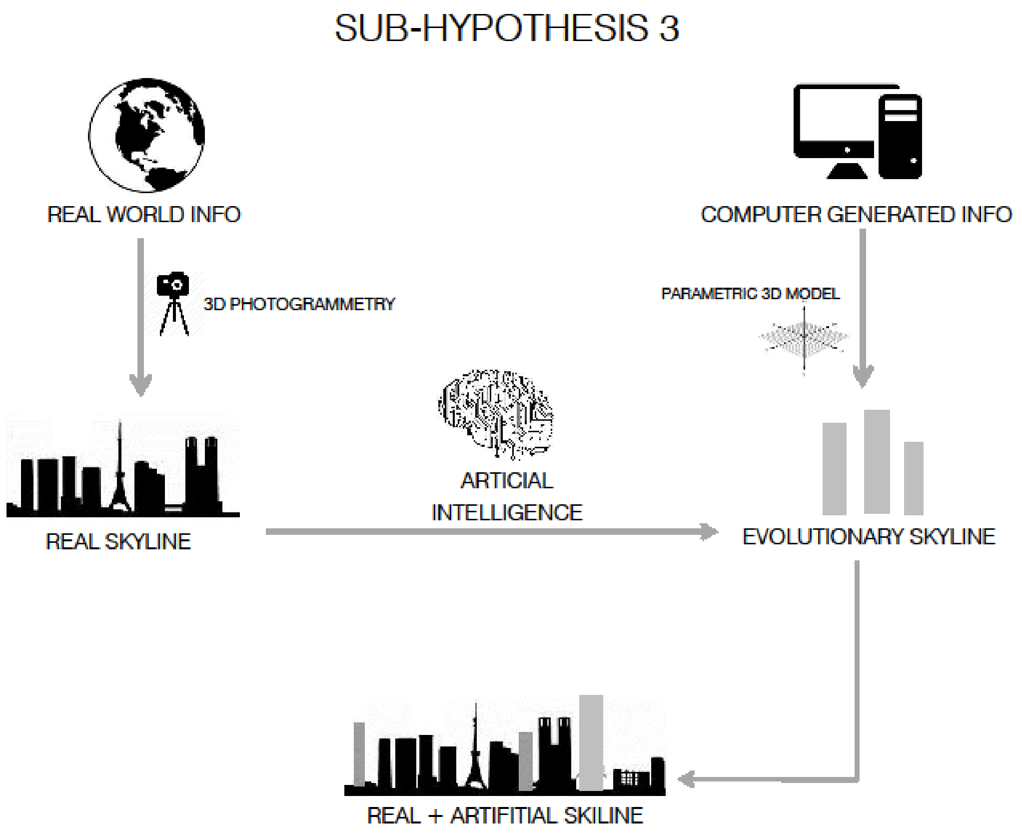


Fig. 49. Diagram of Sub-hypothesis 3 methodology.

The third set of case studies methodological process will follow the next steps (Fig.49):

-An area of study will be defined within a large urban metropolis as well as the building types to be studied.

-Existing parameters that influence the allocation of vertical growth will be analyzed and mapped to later determine the location of possible new buildings.

-A genetic algorithm to predict scenarios of vertical growth will be proposed.

-Economic parameters from the area of study will be used as input data for a genetic algorithm to determine both the number and average height of new skyscrapers.

-A probabilistic allocation map will be generated.

-The genetic algorithm will be tested for the vertical growth of the study area for a 4-year period.

-The results obtained by the genetic algorithm and the probabilistic map will be contrasted with the current observed vertical growth in the area of study.

- The resulting models will combine a ‘real skyline 3D model’, with an ‘artificially generated’ 3D skyline.

-The results will be analyzed and evaluated.

4.2 Overall Methodology.

Once the different case studies are individually evaluated, conclusions will be drawn from the individual results into an overall conclusion to determine whether the combined use of the proposed LIDAR, PBR and evolutionary morphogenesis techniques, results (or not), into more integration between ‘real’ and ‘artificially generated’ 3D models.

By using the Divide and Conquer (D&C) methodological process logic, of dividing a complex problem into simpler sub-problems, individual conclusions will be combined to draw a resulting conclusion.

5 VISUALIZATION OF REAL AND ARTIFICIAL 3D OBJECTS.

5.1 Point-based Rendering of Massive Point Clouds: 'ToView'.

This chapter explores the first sub-hypothesis of the thesis, the use Point clouds formats and LIDAR scanners as a more efficient and accurate way to model and represent 3D architectural and urban objects from the 'real world', to later combine them with 'artificially generated' models, than the most commonly used formats of polygon mesh and NURBS. This chapter will study whether this methodological process could lead to time saving, computation resources savings and most importantly to a better resolution and greater accuracy of 3D renderings.

For such a purpose a software called 'ToView' developed by the Department of Information Technology of A Coruña University will be used as a real-time OpenGL PBR software. The software proposed is able to easily handle point clouds that contain billions of points in real time with a great degree of accuracy and realism.

The software can work with three different visualization algorithms, depending on the desired speed and result. The first visualization algorithm uses image-aligned squares to render points (Fig. 50). This algorithm has the highest performance; splats are rendered using only one OpenGL point for each splat. The next algorithm uses affinely projected point sprites. This algorithm yields a better representation of the splat shape as the one seen in Botsch and Kobbelt (2003). For each of the squares, a 'shader' will discard the part of the square that is outside of the splat. This approach still has some issues when the splats are positioned with extreme angles, which lead to holes in the surfaces.

The last algorithm in the developed software is 'perspectively correct rasterization' of points (Fig. 51). As we mentioned before, the last algorithm had certain problems, as it was not able to correctly rasterize the contour of the splat, leading to holes in the rendered image.

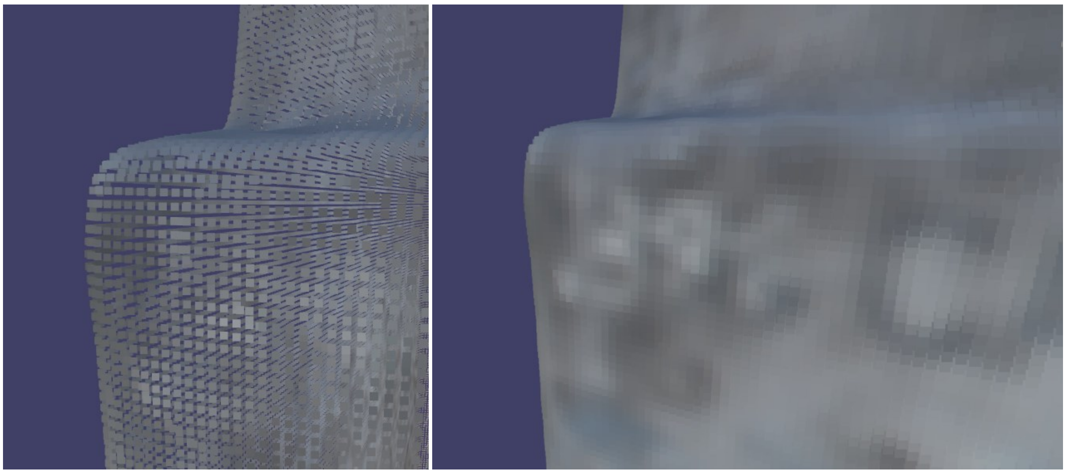


Fig. 50. Rendering technique using image-aligned squares, at different density and sizes.

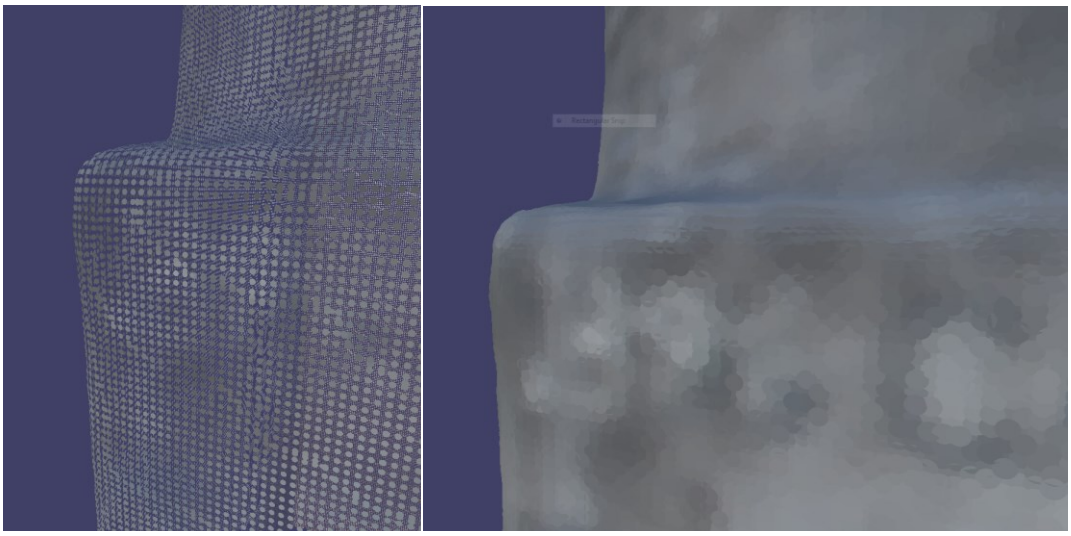


Fig. 51. Rendering technique using perspectively correct rasterization of points, at different density and sizes.

The technique present in Zwicker et al. (2004), transforms correctly the outer contour of the splat, but has problems with the center of the splat. A more efficient and perspective correct algorithm was introduced by Botsch, Spernat and Kobbelt (2004), that has been adapted to fit the needs of the framework. This algorithm was originally designed to represent ellipses; we have modified it to deal with oriented disks, which we will use in our case study (Fig. 52).

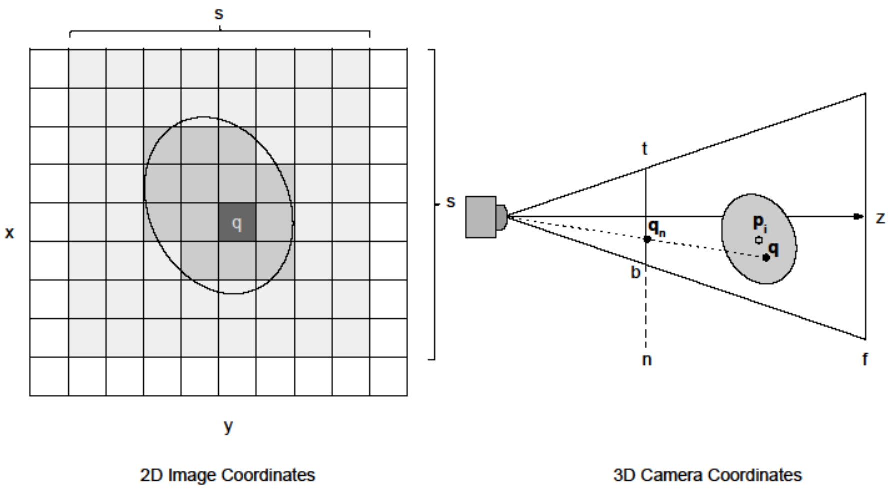


Fig. 52. Perspectively correct rendering.

For cases with irregular point cloud density, 'ToView' is also capable of rendering points of different sizes. This is necessary in order to minimize clipping artifacts and improve rendering quality. The point sizes are obtained in a preprocessing step, in which we can also estimate other point features needed for advanced point rendering. Additionally, to further improve rendering quality, the framework also supports 'multisample anti-aliasing' (MSAA). This technique removes aliasing artifacts and improves surface quality, as can be seen in Akenine et al. (2008).

When dealing with massive point clouds, one of the most important parts of a rendering system is the LOD algorithm, which decreases the complexity of a 3D object representation as it moves away from the viewer. In addition, the multiresolution acceleration structure used is also of the outmost importance. As a result, the final rendering quality and performance is closely related to these two factors. By using an out-of-core KD-tree similar to Goswami et al. (2012) to quickly discard groups of points that are not located inside the camera frustum, it will be possible to render truly massive clouds. Furthermore, the real time renderer also gives higher priority to the points closer to the camera than those far away, since these will have a higher impact in the rendered image.

As the 3D scanned files contain point position (x,y,z), normal and color (illumination level at the time of the laser scan), artificial lighting is not necessary. Since the scanned points already contain illumination values, an artificial lighting model is not necessary to achieve realistic results. In a traditional artificial 3D model, realistic lighting has to be computed using complex global illumination algorithms since no real lighting information is available, yielding less realistic results. Thanks to all of these properties and a complex software cache system, real time rendering of billions of points is possible and smooth, achieving great detail and quality. Given the capabilities of the software and the possibility of rendering real environments in real time with high degree of precision, we will explore and quantify the possibility of using 'ToView' as a software not only for visualizing LIDAR scans, but also as a tool for point-based architectural representation.

5.2 Conversion Algorithm of NURBS and Meshes into Point Clouds.

By using LIDAR scans, it is possible accurately obtain data from existing structures, buildings, spaces or objects from the 'real world' and to merge them with 3D generated designs, basically combining real scanned geometry and manually modeled objects, blurring the boundaries between the real and virtual world. The translation of point clouds into polygon mesh geometry however is not simple and noted on chapter 2. Nowadays, new approaches that employ polygon meshes have partially solved these problems, but still require manual interaction and refinement. This process is usually performed employing a point cloud as a starting point and yields a polygon mesh of the desired object as a result.

The use of PBR as a method of representation is proposed as a more efficient method of representing point clouds than the translation of the latter into polygon meshes, NURBS or other CAD formats. This is especially important when dealing with big environments that contain huge point clouds, in which the manpower and cost needed to convert these datasets to polygon meshes could be easily avoided, resulting in a better representation of these environments. This thesis will explore the use of a conversion algorithm to transform polygon meshes and NURBS, into point clouds as a more efficient way of merging and visualizing scanned 3D data and man-made artificial designs. The proposed algorithm will automate the translation of NURBS and meshes created in Rhinoceros into point clouds with a constant density, in order to facilitate rendering. The algorithm will subdivide each surface into equidistant UV segments, to then proceed with the generation of point equidistant coordinates, including normal, and illumination dependent on the material properties. In order to convert a dataset modelled with NURBS to a point cloud, the surfaces that the model is comprised of have to be sampled and converted to sets of points. Once we have converted all surfaces in the dataset, the process will be complete and it will yield a point cloud that closely represents the mentioned model. Furthermore, that point cloud can then be used to integrate the artificially designed model in a LIDAR scan of a real environment. This can be achieved by employing the visualizer provided in the software framework. When converting a surface to a point cloud, the first step, is computing its domain interval in the u and v directions. For example, the domain of a unit sphere with poles aligned with the Y axis, is obtained using the following equation:

$$u = 0.5 + \frac{\arctan2(d_z, d_x)}{2\pi}$$

$$v = 0.5 - \frac{\arcsin(d_y)}{\pi}$$

Equation 07. domain of a unit sphere with poles aligned with the Y axis

Being d a point in the sphere, the coordinates will yield a value between 0 and 1. To calculate the interval, we need the maximum and minimum u and v values. In most modern design software, this data is easily accessible using their scripting APIs. The next step will be the calculation of the number of points that will be needed to cover the surface. For this purpose, we will calculate how many points per unit (PPU) will be used depending on the needs of the user. Once this quantity is stablished, the number of u_{steps} and v_{steps} is easily computed with the next equation:

$$u_{steps} = (u_{max} - u_{min}) * PPU$$

$$v_{steps} = (v_{max} - v_{min}) * PPU$$

Equation 08. Computation of the number of u_{steps} and v_{steps}.

After computing the number of steps necessary to correctly sample the surface, the next stage will be sampling it. This involves the calculation of the positions and normals of the points that correspond to the samples. In order to accomplish this, the surface will be evaluated in the number of points estimated previously.

For the example sphere, if each u and v coordinateis known, d could be deduced. It will also be its normal, since the used surface is a unit sphere. This step can also be performed using the scripting APIs available in most 3D design software. Furthermore, in some extremely curved surfaces some artifacts can appear. To eliminate these extreme cases, we will discard any point further away from the surface than a threshold.

Once the surface has been completely sampled and the resulting set of points and normal is obtained, the new point cloud has to be artificially illuminated to be able to distinguish all the details. Since flat shading models would not be correctly integrated on a real environment, since the exported models would look flat and lifeless. To artificially illuminate the exported surface, we will use point lights and Phong shading, Bishop and Weimer (1986).

The first step when calculating the color of a point will be the extraction of the material properties of the surface. This can again be obtained using scripting APIs, yielding an ambient, diffuse and specular reflection component. To obtain the ambient reflection of the point is trivial, is the amount of ambient light already present in the material. Next, the amount of diffuse light reflected will be calculated. Since diffuse surfaces reflect light in all directions, the factor that specifies the amount of diffuse light reflected will be:

$$f_d = \frac{l \cdot n}{|l||n|}$$

Equation 09. Amount of diffuse light reflected

Being l the resulting vector from subtracting the light position minus the point position and n the point normal. This factor is then multiplied by the diffuse component and added to the ambient component. The final step needed to obtain the point color, will be the calculation of the amount of specular reflected light present. In order to compute the amount of this type of reflected light, a specular reflection factor is again necessary. The following equation is used to calculate it:

$$f_s = \left(\frac{h \cdot n}{|h||n|}\right)^f$$

Equation 10. Specular reflection factor.

Being n the point normal, f the Phong exponent of the material and h the Blinn halfway vector Blinn, (1977):

$$h = l + v$$

Equation 11. Blinn halfway vector.

Being v the viewing direction of the camera and l the subtraction of the point position minus the light position. This factor is again multiplied by the specular component and added to the other two components, providing our final point color. In order to obtain the best results possible by using the proposed algorithm, the user will choose a light position that most correctly resembles the time of day in which the scan was taken. This will allow the exported cloud to be seamlessly integrated in the desired environment, achieving a more realistic result. The software used to test this new workflow had no trouble importing the resulting clouds, which can be translated, scaled or rotated to integrate these new artificially designed models in real environments. From a practical point of view, the interface algorithm works by importing geometry modeled in Rhinoceros 5.0 into 'ToView'. The next paragraphs show the workflow in a practical example.

5.3 Real Time Rendering of LIDAR Scans and 3D Geometry.

In order to test the proposed workflow, first will be necessary to manually model a 3D geometry, add a color and add a light to match the light from the 3D LIDAR Scan, to be later merged into a scanned real world environment. In this case a very simple house was manually modeled to test the algorithm, Fig. 53.

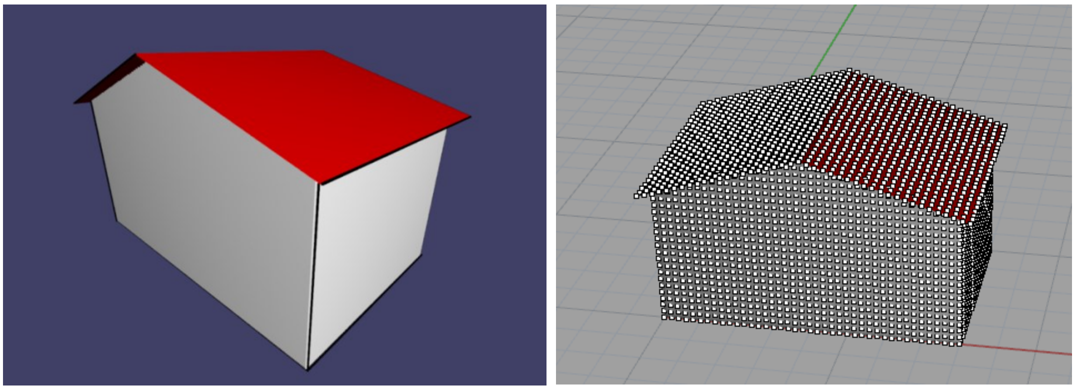


Fig. 53. On the left, simple house geometry manually modelled in Rhinoceros 5.0. On the right, conversion to an equidistant point cloud.

The second step in Rhinoceros is to load the script under the Tools menu, on the RhinoScrip tab select 'Edit'. Then choose the script previously developed from the menu. Then follow the steps, and choose the point density. The script will automatically convert the NURBS geometry into an equidistant point cloud (Fig. 53). The point cloud will then be imported into 'ToView' and placed into the right location by using rotation and translation commands.

The point geometry is rendered in 'ToView' successfully combining data from a LIDAR scan and a house manually modeled in Rhinoceros. As shown in Fig. 54 even if the algorithm succeeded, there is a clear visual difference between the object modeled manually and the 3D LIDAR scan data.

Once the architectural data is translated into a point cloud and imported into the PBR software, in this case 'ToView', this artificially generated data will be easily merged with scanned 3D data with millions of points obtained from the real world. As previously noted, the point data already contain illumination information, for that purpose it is important to previously match the light levels and direction of the 'artificially generated' model to those at the time of the 'real world' scan.

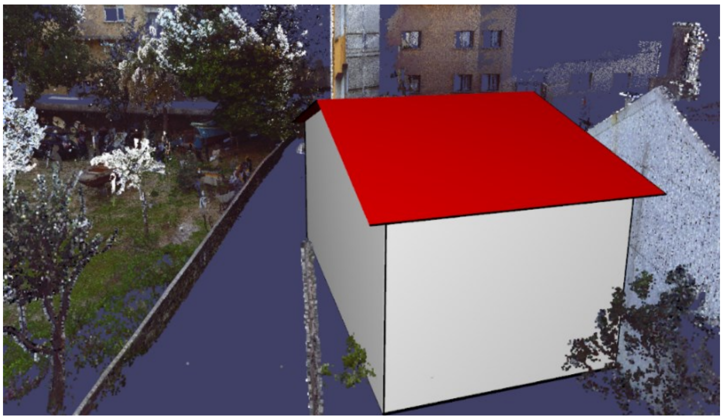


Fig. 54. PBR render of the point cloud 3D models.

To further evaluate the proposed methodology, and to solve the issues experimented on the previous paragraphs, a more detailed example has been developed. A real historical urban environment has been scanned using LIDAR technology as a high density unstructured point cloud and rendered in the PBR software 'ToView', Fig. 55 and 56. For this case study the historical a historical urban setting has been chosen, in particular the Baron Street Crossing, a small medieval urban environment in the city of Pontevedra, in Northwestern Spain.

Fig. 55. PBR 3D visualization of a real urban environment, previously scanned using LIDAR technology.

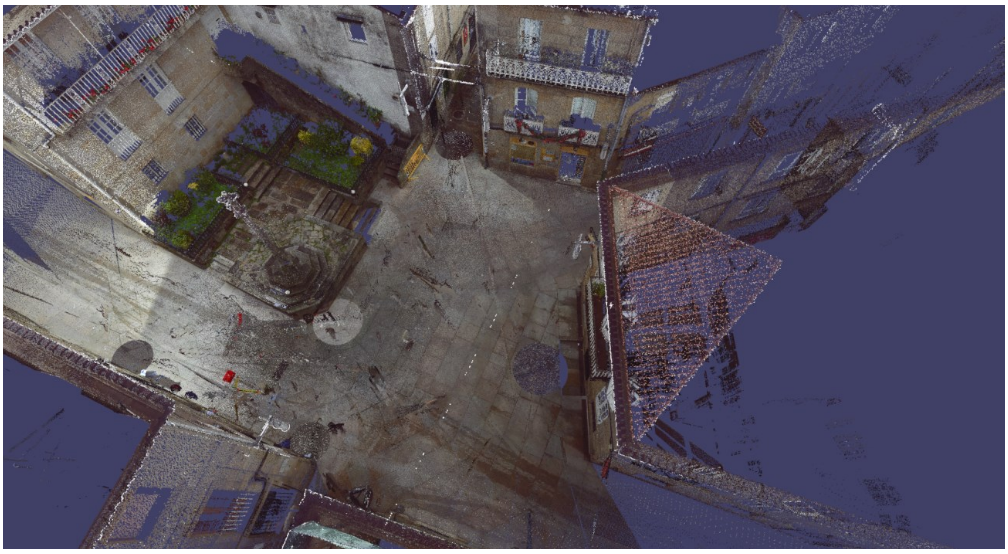


Fig. 56. PBR 3D visualization of a real urban environment, previously scanned using LIDAR technology.

The areas shown in blue on Fig. 55 and Fig. 56, represent the background or empty areas, basically areas where the laser beam was not able to reach, leaving a hole (or blank) into the point cloud. In order to fix this limitation of the Laser scanning, the missing areas can be fixed manually in the 3D point model, or the renderings can be post-processed using an image editing software, this last option was chosen as it faster and yields better visual results.

The final output of the 'ToView' PBR visualization are very realistic, almost at photographic level after post-processing (Fig. 57).

Fig. 57. Raw LiDAR scan rendered in 'ToView'. Real time open GL screen capture, with post processing. 'Calle Baron' in Pontevedra, Spain.

The next step has been to generate a manually modeled artificial 3D geometry to be merged into the previous scan. A schematic conceptual 3D model of a new building has been manually made in Rhinoceros 5.0, as the software allows for complex modeling and supports many different formats as well as scripts. The original NURBS model was then converted into point geometry by the conversion algorithm previously developed and exported as a point cloud (Fig. 58). Currently no other software is able to do this kind of operation, except for Rhinoceros 5.0 combined with the script developed. The algorithm created has been tested and implemented, as a plug-in order to import the 3D geometry as a point cloud in ASC format, as both Rhinoceros and 'ToView' are able to work with this format. The suggested algorithm works by presetting a specific point density and illumination levels to any surface. The ASC file was imported into 'ToView' and manually placed into the desired position.

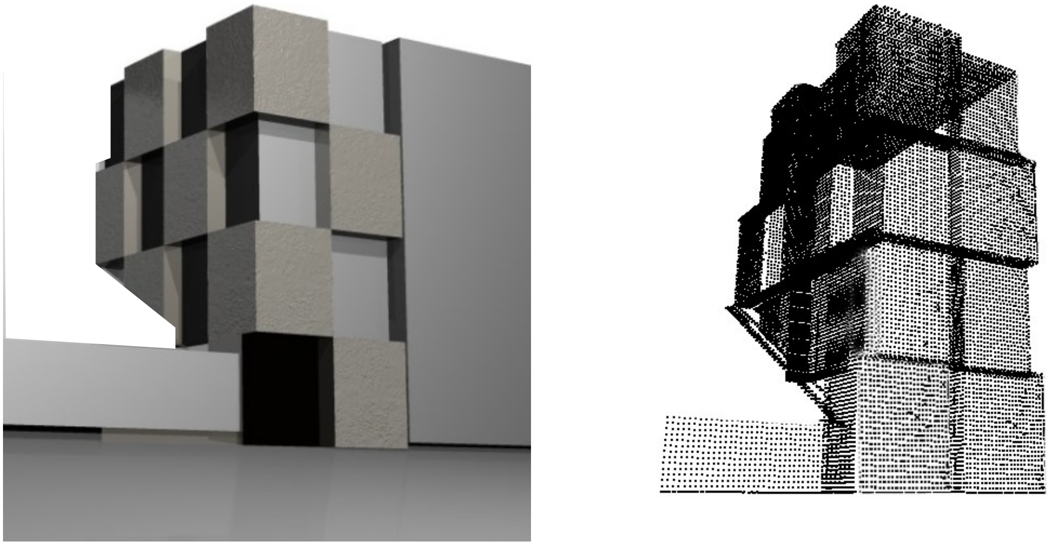


Fig. 58. On the left conceptual Building design, NURBS modeled. On the right the 3D Model exported as a point cloud by the conversion algorithm.

Fig. 59. Post-processed PBR rendering of the LIDAR scanned urban environment. Highlighted in red the area that will be manually modified.

Fig. 60. LIDAR scan rendered in 'ToView', merged with the artificially generated. Post-processed PBR Real time open GL screen capture.

By the implementation of 'ToView' and the conversion algorithm it is possible to visualize together LIDAR scanned objects with architectural designs, and visualize them in real time as a point cloud (Figs. 59 and 60). The proposed methodology not only allows for a smooth workflow and takes full advantage of the capabilities of LIDAR technology, but also achieves more realistic results and faster processing than non-point-based renderings engines. Allowing not only for real time rendering of laser scans, but also for the integration of manually modelled 3D formats and further manipulation of the scanned data. Using point geometry and PBR processing is much faster, and the results are more accurate in terms of geometry, color, textures and light levels than manually modelled polygons or NURBS formats.

5.4 Digital Modeling and Visualization of Complex Geometry.

In order to further explore the proposed methodology and additional example has been tested, this time a smaller object of very complex curve geometry was chosen.

This third study will explore PBR, LIDAR scanning and manual modeling of complex curvature geometries. An example of pre-computer representation of complex geometries has been chosen, in order to further develop it, but this time with the aid of computation techniques. The research starting point is an essay by architect Enric Miralles, in El Croquis magazine, titled 'How to layout a croissant', Miralles and Prats (1991). The article explores the representation of an existing quotidian small object of complex geometry, in particular a croissant. Drawing a croissant was indeed a very challenging exercise at that time, as its geometry is made of irregular complex curved surfaces that resulted from a baking process, making all croissants similar, but actually none of them identical. Accurate Cartesian representation of its plans and sections, and in particular its dimensioning is not as straight forward process and inquiries into lots of inaccuracies. Miralles and Prats (1991) suggested to measure the plan over a system of triangles (which are non-deformable geometry), then to draw the sections, focusing mostly on the straight tangential segments, to later define the curvature radius and allocating its centers, layout the position of the sections, and more specifically the centers of curvature of the tangential radius (Fig. 61). They rationalized the croissant geometry by accurately defining the curvature radius and tangents, transforming a set of continuous irregular surfaces into lines and arcs. Their purpose was to translate a highly irregular geometry into a rational one. It is very obvious however that the croissant drawn by them, even if it has a very similar geometry than the original one, it is intrinsically different, as the rationalization lacks the imperfections of the original geometry.

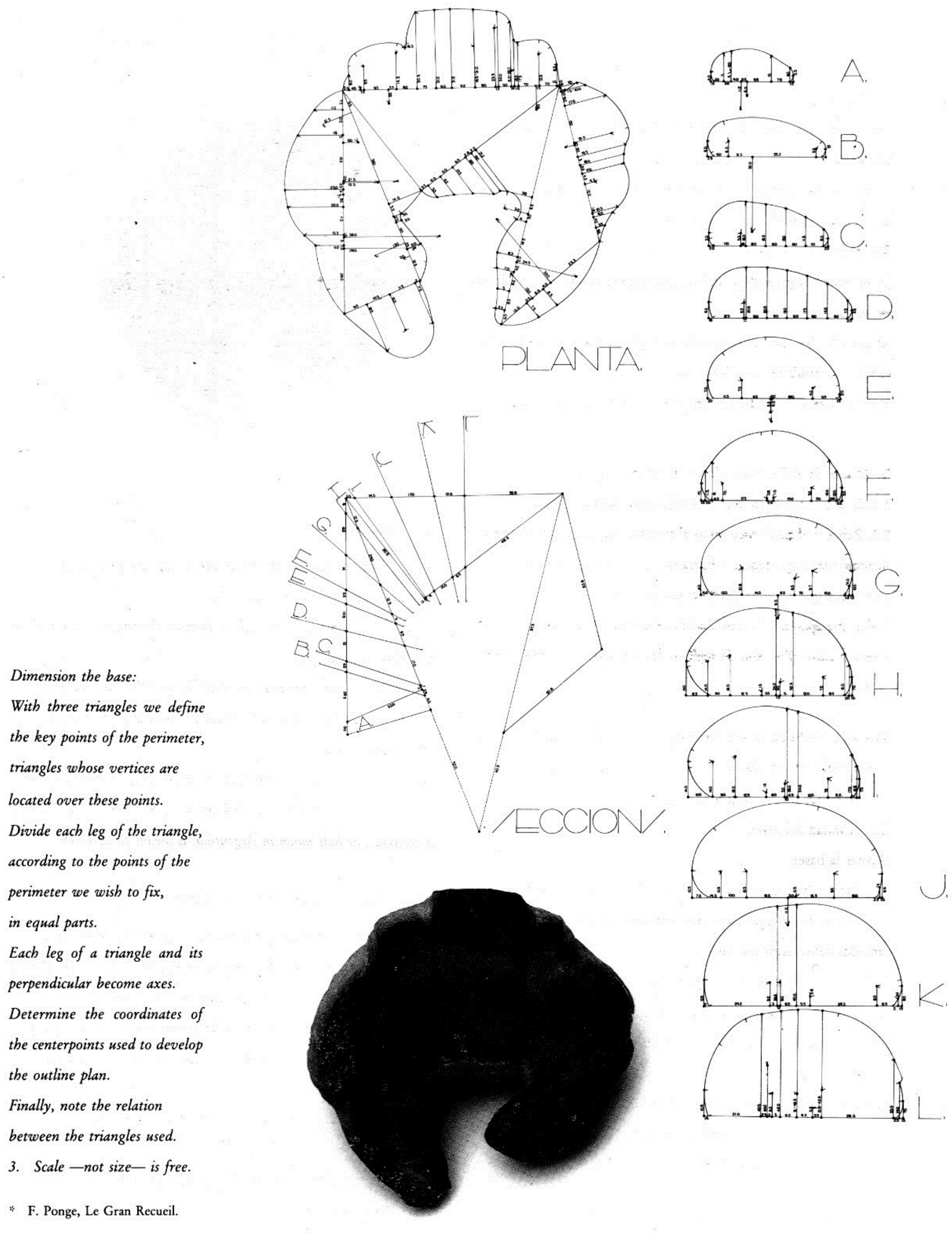


Fig. 61. Miralles and Prats article 'How to layout a croissant', explores the representation of geometrical complex objects. Miralles and Prats (1991).

The croissant will be used as example to further investigate the geometrical and representational challenges faced by Miralles and Prats in order to then easily compare pre-computer and post-computer representation, but any other complex curved object of complex curvature could have been used instead. Different 3D formats and software will be used in order to model and visualize a croissant to then evaluate the results. First some of the most common software packages for architectural design will be used, such as CAD, Rhinoceros and 3DSMax. For such a purpose a croissant was first sliced and manually measured (Fig. 62), to then generate plans and sections as accurate as possible, the 2D orthogonal projections were then drawn using CAD software (AutoCAD 2014).

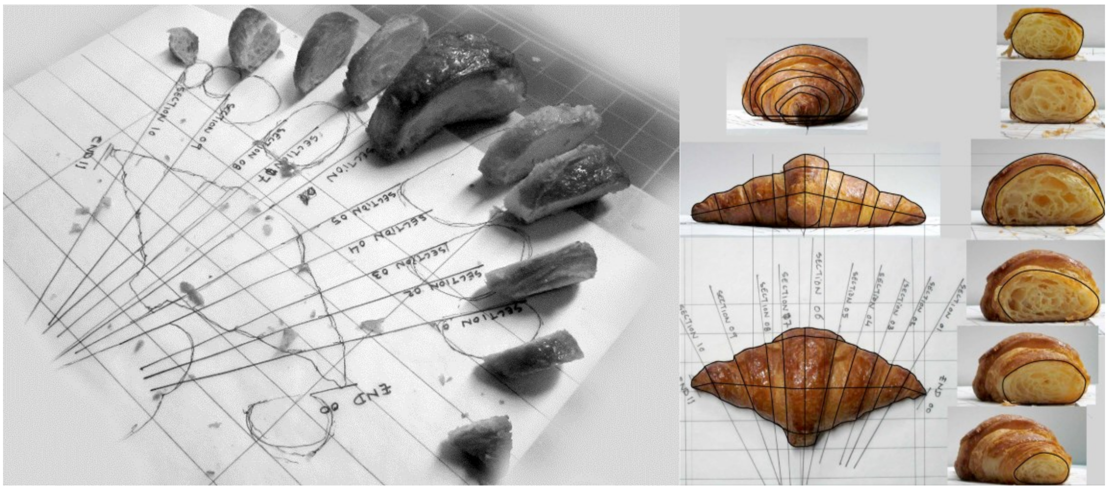


Fig. 62. Manual slicing and measurement process.

The CAD sections and plans were then exported into Rhinoceros 5.0 and modeled as NURBS, as the software allows for very accurate manipulation of curved geometry, as the one on a croissant surface. The NURBS splines were lofted in order to create a smooth continuous surface.

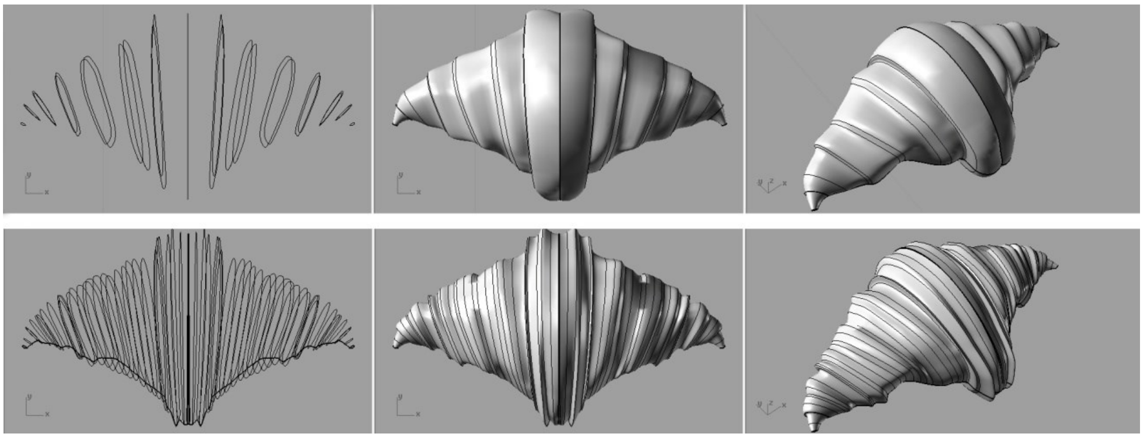


Fig. 63. The top row shows the result of manually modeling the croissant. The bottom row shows additional manual manipulation of the surfaces to achieve a more convincing representation.

As a result of the loft process from the original sections, the geometry obtained was convincing, but it lacked all the irregularities and imperfections found on a real croissant, for that purpose the croissant geometry was further modeled manually, by looking at the pictures of the original croissant, in order to achieve a more realistic and closer result to the original geometry (Fig. 63). By doing this manual process, the model has become more realistic, however further differs from the original one.

To obtain a realistic representation of the croissant, the model was converted into a polygon mesh, and then rendered in Autodesk 3DSMax (Fig. 64), using the rendering engine from The Chaos group VRAY, Kuhlo and Eggert, (2010). The combination of both software packages achieves some of the most realistic results for architectural rendering.

Fig. 64. 3D renderings of the croissants. The figure on the left shows the lofted 3D croissant model. The image on the right shows the additional manual texturing on the croissant's surface.

Furthermore, in order to achieve an even more irregular surface and realistic color and texture, a jpg map, from pictures taken from the original croissant was applied to the diffuse, texture and bump channels, using UVW cylindrical mapping (Fig. 65).

Fig. 65. 3DSMAX and VRAY croissants, adding jpg diffuse, bump mapping and reflection

The rendered croissants even if quite realistic differ substantially from the original one, both in geometry and mapping. As the new geometry is just a manual 3D approximation of the geometry of the original croissant. The mapping is the texture from the original croissant, but the pixels of the map do not fall in the exact same

locations of the original sample (Fig. 65), are actually the same croissant. By using polygon mesh and NURBS geometry, results with a high degree of accuracy have been achieved. However, the geometry generated even if very similar still differs in +/-2mm from the original sample on the entire surface. Also, the croissants lack of the natural imperfections of the original one, which were manually modeled.
In order to obtain more accurate measurements, the use of a LIDAR scanner was tested, by obtaining a replica of the original geometry of the croissant as a point cloud format.

5.5 LIDAR and Point-based Visualization of Complex Curved Geometry.

A different croissant has been scanned by using LIDAR 3D scanning technology (Fig. 66). LIDAR laser scanners, as previously mentioned, use massive point clouds as output data, and the information obtained from the scans is an unstructured massive point cloud, basically extensive databases of unconnected points, without a specific structure or order.

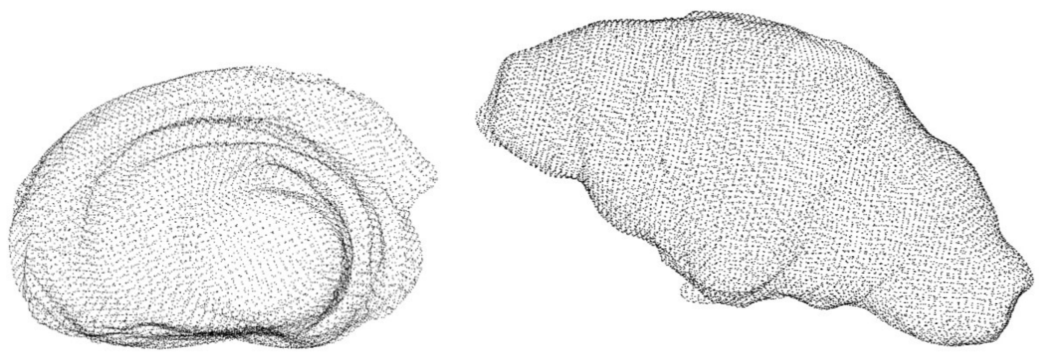


Fig. 66. Unstructured point cloud geometry, resulted from the LiDAR laser scan.

The real-time OpenGL PBR software 'ToView' will be used again. As tested in the previous example 'ToView' is able to easily handle point clouds with billions of points, which regular PCs are able to handle in real time with a great degree of accuracy and realism.

LIDAR scanning and PBR results in an accurate 3D visualization in real time of the croissant, to a level of precision of up to half a millimeter, in that sense some of the original issues have been partially solved (Fig. 67). The surface now also has the natural imperfections that were lacking on the manually generated 3D model.

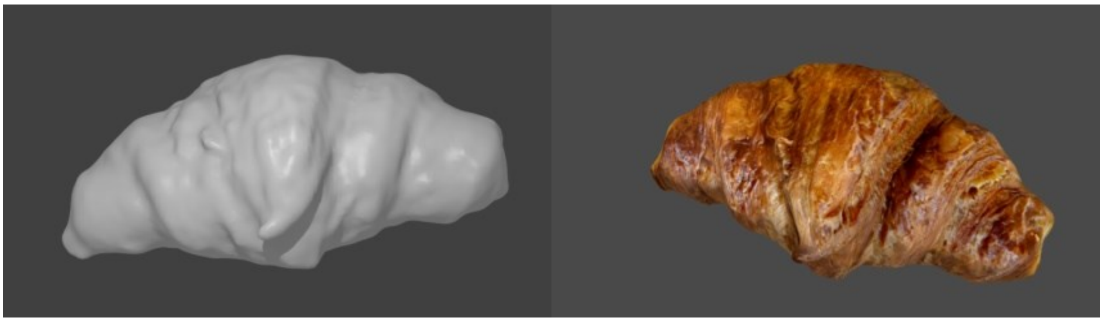


Fig. 67. PBR of the geometry obtained from a LiDAR scan of a croissant in 'ToView'.

LIDAR Laser scanning has its own limitations, as certain areas are difficult to scan, as its geometry it is hidden to the laser beam (shadow), reflections of the laser bean might result into inaccuracies and other minor issue such as noise might happen. The rendering results using the PBR software 'ToView' are satisfactory, with a visual quality similar to those generated by the manual NURBS modeling and VRay rendering (Fig. 68).

By using LIDAR scanning and PBR a much closer result to the original sample has been achieved. In that sense point-based geometry and representation resulted into a more accurate representation than polygon meshes. On the other hand, more visually compelling results have been achieved with the polygon mesh rendering in 3DSMax and VRay, due to the fact that the software packages are more developed and allow for great visual effects and enhancements, in many cases improving the quality of the original real objects and environments (Fig. 68).

It is not the final visual effect what is being evaluated, but the geometry itself.

Fig. 68. On the left unstructured point cloud from a LIDAR scan rendered with VRay. On the right the manually modeled polygon mesh rendered in 3DSMax with VRAY.

5.6 Results.

Chapter 5 explores sub-hypothesis 01, and shows how PBR in combination with the conversion algorithm proposed allows for an accurate and smooth integration of objects from the 'real world' and 'artificially generated' objects.

Points can be used as an aid for the representation of vectors, surface edges and intersections, but also, they can be a very efficient tool for architectural representation, by using topological geometries and PBR visualization.

PBR is a novel computational technique, but the idea of using points as a method of representation was suggested as early as the sixteenth century by Dürer. Points have always been the underlying structure of architectural representation, and they had particular importance for the representation of complex geometry and descriptive geometry. The interpolation of points is the generating structure of vectors, polygon meshes and NURBS. However, point clouds have not typically being used for final output for architectural representation, but just as a generation method. The novelty about PBR is the fact that point clouds don't have any connectivity information, in that way computation of topological systems is more efficient than other geometries.

The Case Studies tested on chapter 5, have successfully proven that the PBR software ‘To View’ developed by our research group combined with the proposed conversion algorithm, allows for efficient rendering. It also demonstrates that it has successfully combined 3D models obtained by directly scanning from the ‘real world’ with ‘artificially generated’ models, by obtaining more accurate and realistic results.

6 EVOLUTIONARY MORPHOGENESIS OF REAL AND ARTIFICIAL 3D’s.

6.1 Genetic Morphogenesis. Surface Texturing Algorithm: ‘Tadpole’.

This chapter will test the second sub-hypothesis, which is the use of artificial intelligence techniques to achieve a more realistically result, in order to resemble objects from the ‘real world’, and more in particular to create diversity and complexity (imperfections), as found in natural objects. The use of artificial intelligence techniques will also assist in the design process of generation of new 3D artificial objects and designs, by the use of computational design techniques that given a problem will be able to auto generate, select and recombine a great number of design solutions.

A genetic algorithm is proposed, Holland (1975); Goldberg (1989), 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 today is widespread, Goldberg (1989); Bentley (1996).

The novelty of the use of this algorithm is that it uses genetic codes in order to create a natural texture from a point cloud representation of the original surface. Unlike Rhinoceros, Grasshopper and other parametric software packages, it is not the user who decides how the surface gets deformed, but the code. In fact, the user simply decides which of the examples offered by the system are consistent with his/her criteria, just as design evaluators, Dawkins (1987), Todd and Latham, (1992). This interaction between human and computer has some advantages especially in the creative process. The user chooses following the criteria it deems appropriate without ‘wasting’ time in the execution process, which tends to be costlier in terms of time.

The algorithm is based on interactive evolutionary computation (IEC), 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, Gero (1999). 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.

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.

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 PPU are used is carried out. Once this quantity is established the number of Usteps and Vsteps is easily computed with the next equations:

$$u_{steps} = (u_{max} - u_{min}) * PPU$$
$$v_{steps} = (v_{max} - v_{min}) * PPU$$

Equation 12. Computation of the number of u_{steps} and v_{steps}.

For 'polysurfaces', the Usteps and Vsteps 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, Rovira et al (2005). (Fig. 69)

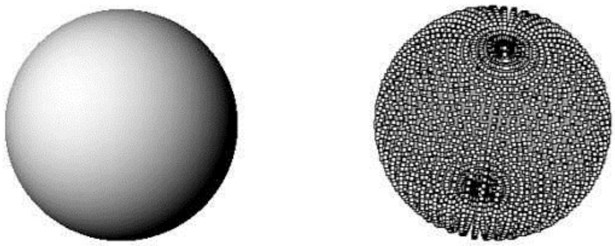


Fig. 69. Point cloud sampling from sphere.

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 IEC, Dawkins (1987), 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, Gero (1999). The user evaluates the individuals guiding the evolution process.

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 (2000, 2002), 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 Machado and Cardoso (2000).

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. 70).

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.

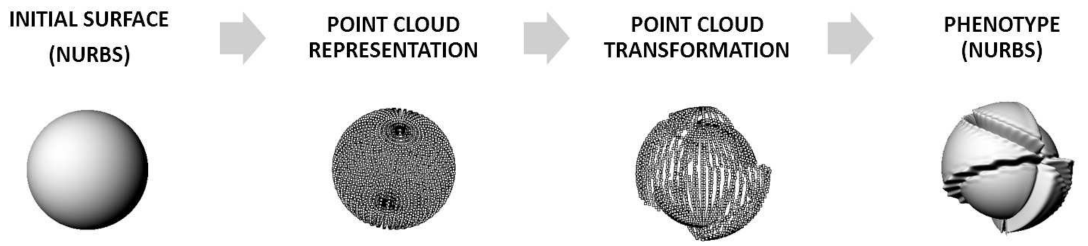


Fig. 70. Visual genetic morphogenesis process.

The algorithm 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' Z0. It is represented with Zf 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 (Z0, parents): computes reproduction and mutation using both parents and the initial surface Z0.

-IsValid (offspring): checks if offspring fulfill all design preconditions / constraints.

-Insert (offspring, Q): this function first converts the offspring genotype to its phenotype and after is added to the auxiliary population Q.

-User evaluation(): once $|Q| = N$, the user will determine 'deletable' designs keeping those which will participate in next step.

Looking for the difference of the proposed algorithm, it has to be noticed the possibility of "Hermaphroditism" allowed in the roulette wheel selection method. 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 (pm) and crossover (pc) vary depending on the percentage of designs that survive each iteration and the range of allowable values for pm.

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

$$p_c = 1 - p_m$$

Equation 13. percentages of mutations and crossover.

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.

```
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
```

Table 01. Algorithm GP pseudocode.

Unlike DeJong's (1975) method, 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 pm 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 02. 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.

Parameter	Value
Point-cloud UV density	50 by 50 pps
Range of displacements	[-0.1 ; 0.1] cm
Minimun Affected surface	45%
Maximum Tree Size	40
Minimum Tree Size	5
Functions (Tree Nodes)	+, -, *, /
Terminals (Tree Leaves)	"x", "y"
Chromosome encoding	Tree Structure
Initialization	Ramped half-and-half
Fitness function	Binary EIC
Recombination Strategy	1-point crossover
Mutation Strategy	Leave-flipping
Mutation rate (p_m)	[0.01; 0.15]
Selection Strategy	Proportional Roulette Wheel
Replacement Strategy	Invert-fitness

Table 02. Predefined parameters used by the GP approach for the design process.

Due to this, three graphical representations corresponding to Matyas, SixHump Camel and Easom functions, Molga and Smutnicki (2005), in the range [-1, 1] are used as goals (Fig. 71). These functions are well known and widely used in optimization.

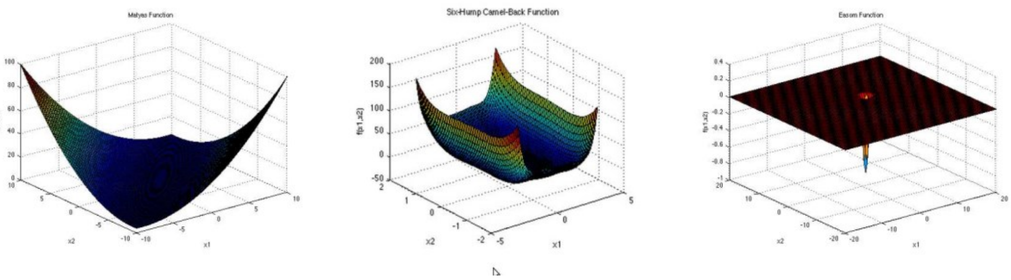


Fig. 71: Optimization functions used for convergence testing.(Left) Matyas (Plate-Shaped), (center) SixHump (Valley-Shaped), (right) Easom (Steep Ridges).

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 ‘Sx’ with regard to the representation in point cloud of the target surface S0 using both mean and standard deviation of every point in the cloud simultaneously. The fitness function exponentially penalizes 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 minimized. To confirm the stochastic stability of the convergence, 50 independent runs for each objective surface has been carried out. Fig. 72 details the average fitness of the best individual. Results confirm the convergence achieved with a final fitness < 0.005.

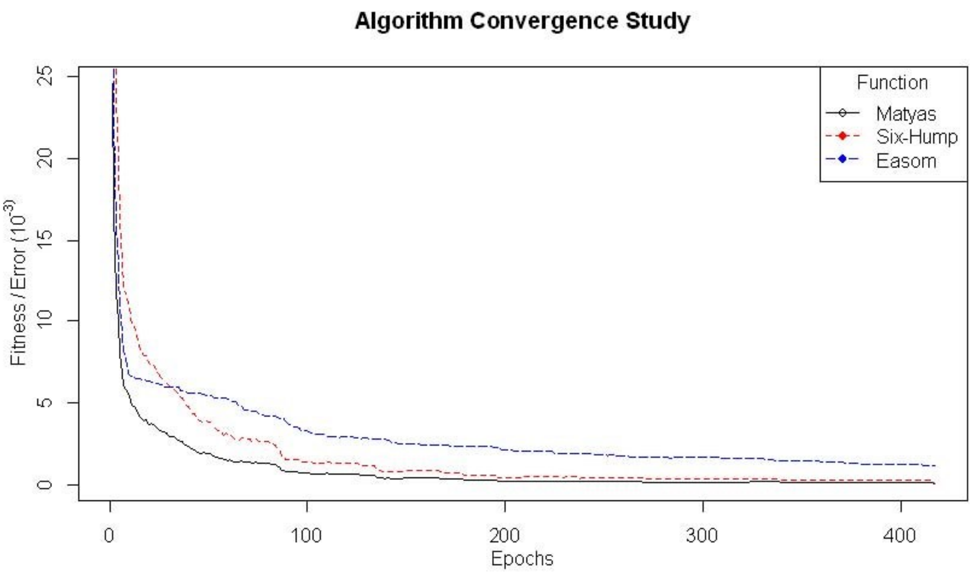


Fig. 72. Convergence graph for the 3 objective functions.

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. 73 represents 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.

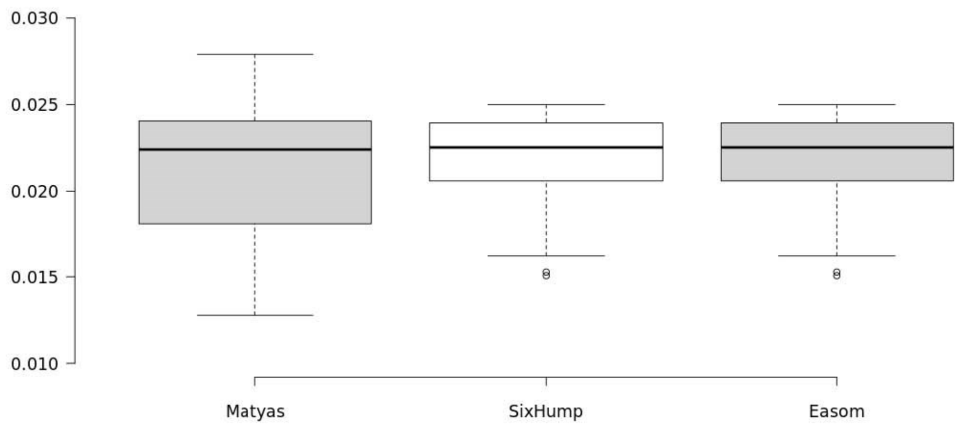


Fig. 73. Diversity boxplot.

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(nlog(n)), noting that performing point cloud with sizes over 10000 points in home computers is untreatable. However, the complexity attending the population size if O(log(n)). Hence, the algorithm complexity is O(nlog(n)).

6.2 Surface Morphogenesis of Complex Curved Geometry.

As in the last example of Chapter 5, for the first surface morphogenesis case study a croissant will be used, in order to test the generation of deformations on its surface, and in particular to be able to generate diversity, by creating families resulting from the manipulation of the original shape. The objective of this case study is to artificially generate a croissant 3D model with surface imperfections and to auto-generate diversity as in a real batch of croissants.

For such a purpose a topological evolutionary morphogenesis process is proposed. In order to introduce the imperfections on the croissant surface and to generate a diversity of similar solutions, the genetic algorithm developed on the previous sub-chapter will be used. The idea is that from the point-based geometry of an existing croissant as a starting point, to generate the imprecisions on its surface not manually but by a morphogenesis process. In addition to automatically achieve diversity from an original 3D sample, (Fig. 74).

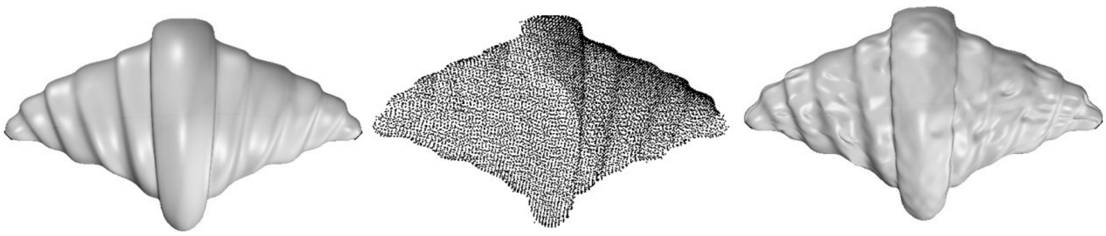


Fig. 74. Croissant representation transformation that take place during the algorithm application. (From left to right: rendered geometry, point-could modification and re-rendering).

A series of initially preset parameters will determine how the deformations are applied, such as: point cloud density, range of min/max of displacements, minimum percentage of affected surface, maximum and minimum genotype's tree size of the first population and those related to the evolution process (mutation and crossover rates). All of these parameters used in the following case study are detailed in Table 03.

A basic canonical GA is used with roulette wheel partner selection, Guilford (1967). For the recombination, a standard GP crossover operator is used and a basic mutation operator for both nodes and leaves, Koza (1992).

PARAMETER	*VALUE*
Minimum percentage of affected surface	95%
Minimum Tree Size	15
Crossover rate	85%
Mutation rate	15%
Functions (Tree Nodes)	+, -, *, /, %
Terminals (Tree Leaves)	“x”, “y” coordinates

Table 03. Parameters used by the algorithm predefined for the design process.

Through a process of manual selection of those textures more similar to those in the original object (in this case a croissant) some clones will survive, and that surviving mathematical code will produce descendants. The algorithm will then use their generic formulations to produce additional singular clones and so on till the user considers necessary.

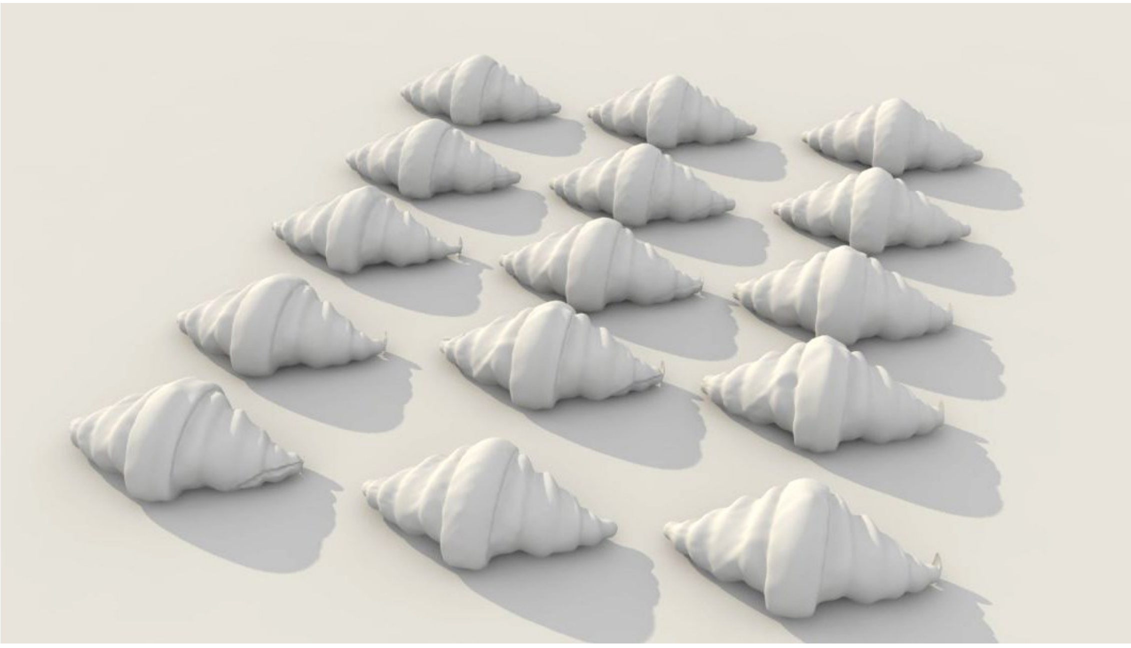


Fig. 75. Croissant resulted from the genetic morphogenesis algorithm. Diversity has been achieved as each mutant surface is partially different to each other.

By using this algorithm, not only achieved random deformations of the surface have been achieved, but also an infinite diversity of similar solutions (Fig. 75). It is important to note that the proposed algorithm works with point clouds. By translating the NURBS geometry first into a point cloud, then applying a genetically-based mathematical formulation over all points that then is rebuilt into NURBS geometry. The use of point geometry to generate the first sample of the morphogenesis process resulted into a more accurate representation of the original sample.

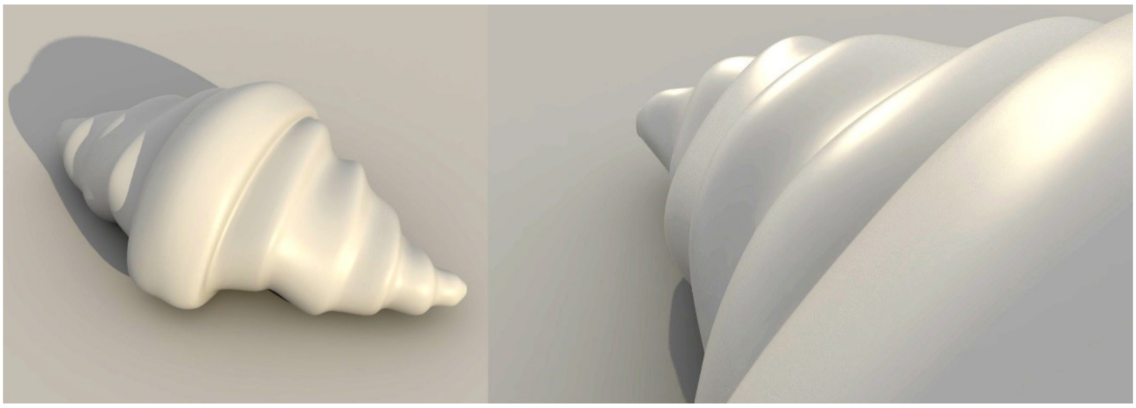


Fig. 76. Original manually generated rationalized Croissant 3D model.

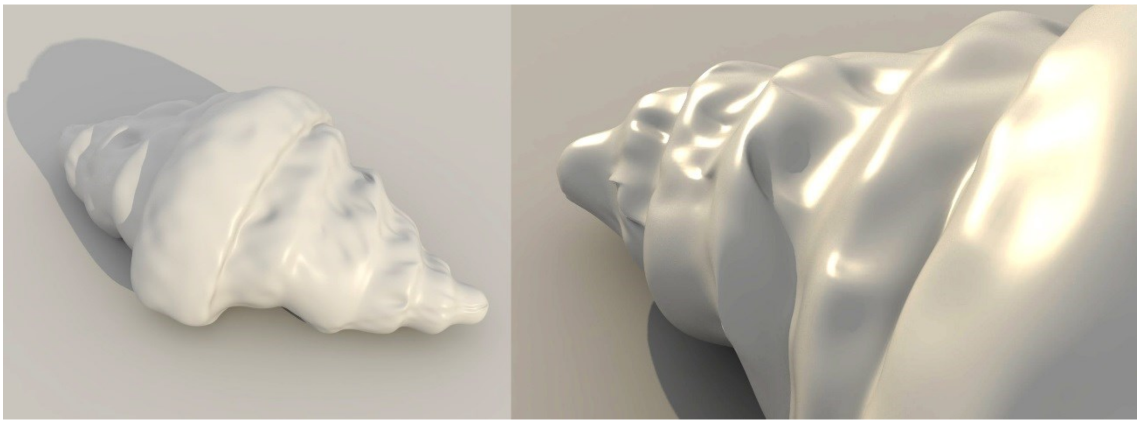


Fig. 77. Croissant 3D model surface resulted after the evolutionary morphogenesis process.

By the using a genetic evolutionary morphogenesis algorithm some of the initial challenges have been solved, resulting into the generation of diversity by producing irregularity on the surfaces, furthermore it has been achieved through an evolutionary process (Fig. 76 and 77). The unique characteristic of the process of modeling is that it is possible to modify and evolve the unstructured point cloud by a genetic recombination of surviving samples, not by manually manipulating the parameters.

6.3 Morphogenesis of Architectural Surfaces.

Next case study will apply the methodology of the previous case study for the generation of a complex 3D architectural surface. The case study combines the translation of NURBS and meshes into point clouds for representation and morphogenesis purposes.

It will explore the use of a GA not just as a 3D modeling tool, but as a tool for the generation of designs, producing numerous related samples driving the morphogenesis process. The algorithm has been tested to first produce an industrially designed object to be later mass produced, in particular a cast ceramic tile with an irregular surface. This kind of tiles are common for architectural applications (Fig. 78), we can find numerous examples in recent contemporary architecture.

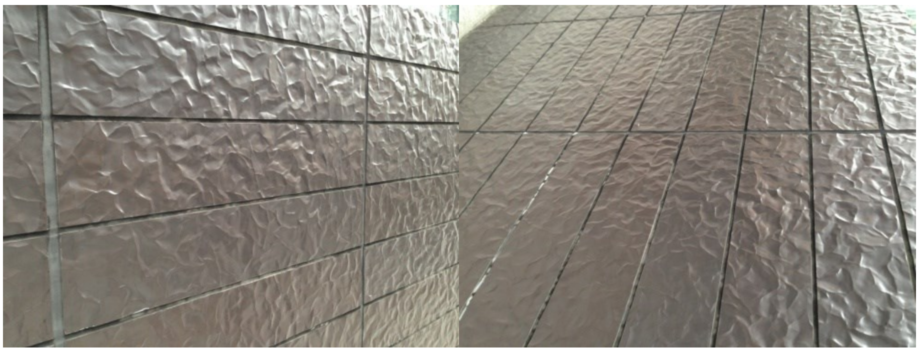


Fig. 78. Real example of an application of irregular surface tiles, on a building in Tokyo.

By applying the GA to a flat NURBS surface, families of tiles were achieved, all sharing similar irregular 3D texturing but all different (Fig. 79 and 80). The novelty about this process is that what drove the final result has been the algorithm by a parametric and evolutionary process, and in particular the genetically selection of successful designs in order to naturally lead to a final design result. The Designer instead of drawing will run the system by modifying the parameters, to later select and recombine the successful geometries following an evolutionary process.

Fig. 79. Close up of the tile design resulting from the GA, evolutionary selection process, by matching successful samples into generating new ones.

Fig. 80. Diverse batch of tiles generated through an evolutionary process.

6.4 Surface Evolutionary Morphogenesis of a High-Rise Building.

The previous design methodological processes will be used again for a third morphogenesis case study, this time to design the entire massing of a building. The 265m tall Beekman Tower (Fig. 81), designed by Architect Frank Gehry in New York

City has been used as a starting point for this case study. The high-rise apartment building has an exterior envelope made as a continue morphing surface of stainless steel panels. According to Sundberg (2011): 'SolidWorks was used for all-important parametric modeling and CATIA, a 3D modeling program widely used in the automotive and aerospace industries to realize his unorthodox undulating shapes, was employed for surfacing'.

Fig. 81. Beekman Tower, by Frank Gehry Partners LLP, completed in 2010. (Photo: Gehry Partners Source: architizer.com).

The GA previously created to be applied to the tiles and to the croissant has been tested in order to generate a similar type of design, but this time by applying an evolutionary methodological process. The GA has been applied to a flat surface in order to generate an experimental skyscraper façade design, using similar parameters than the tiles design. For such a purpose the deformations parameters have been increased, until some desirable effects have been achieved. The algorithm have assisted not only to produce the deformations itself by modifying the parameters, but also into an selective design process, by producing more design options based on our design deformations picks, eventually generating hundreds of families of design results, leaving to us to 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 was been selected (Fig. 82), and then in order to produce a final design of a building a simple pattern of windows was projected into it.

Fig. 82. Design process of making windows and rendering over the selected auto-generated surface.

On purpose deformations that will produce a similar design effect than Gehry's Beekman Tower were selected, in that sense a combination of parametric design and evolutionary selection has driven the process resulting in a similar design (Fig. 83).

6.5 Results.

The evolutionary morphogenesis algorithm has proven successful for both the generation of irregularity and diversity from an original sample. The GA morphogenesis algorithm has been yielded satisfactory results when tested for the generation of irregularities and diversity in surfaces, through three different case studies: a complex curved object, an architectural tile and a building's facade. In that sense the GA morphogenesis algorithm has proven itself successfully not only for the representation but also for the generation of morphologies through evolutionary selection to be later implemented as final designs.

Fig. 83. Tower design resembling the Beekman Tower, this time generated by the GA algorithm, and rendered in VRay.

The unique characteristic of this process of modeling is the user's ability to modify and evolve the point cloud by a genetic recombination of surviving samples.Two objectives have been achieved, the first to artificially generate more realistic 3D models using artificial intelligence. The second to obtain more realistic imperfections and diversity as in the 'real world'. The proposed process not only auto-generates designs, but also generates more realistic outputs that if the same processes have been done manually. Not only a greater level of reality has been achieved, but also the algorithm was able to auto-generate new designs based on pre-stablished parameters.

7 EVOLUTIONARY VERTICAL URBAN GROWTH.

7.1 Introduction.

This chapter explores the use of AI techniques to simulate the growth and evolution of complex real urban geometries (sub-hypothesis 3). Human-made complex self-organized systems such as cities are in constant evolution. Evolutionary computation will be used to auto-generate, evolve and predict the evolution of self-organized systems, merging the boundaries between 'real urban growth' and 'simulated urban growth'.

This chapter will propose a GA, to be applied to a 3D model of a city to simulate different scenarios of growth. The GA will be based in pre-established parameters, to auto-generate possible scenarios of growth of a real geometry, blurring the boundaries between 'real growth' and 'artificial growth', and between present and future.

Tokyo will be used as case study, for several reasons. Tokyo is the largest continuous metropolitan area in the world, the third tallest per the CTBUH (2009) and one of the fastest growing skylines in recent years (Fig. 84). As previously explained in the Fundamentals section (2.2.4), many authors have linked the growth of cities to biological growth, as the patterns of development of cities typically resemble the patterns of growth of organisms, governed by both evolutionary and self-organizing processes (Al Sayed et Turner 2012). This doctoral thesis proposes an evolutionary algorithm, based on previously identified recurring morphological patterns. The proposed GA when applied to a 3D model of a city's geometry will replicate through morphogenesis processes its vertical growth. The parameters will be generated from data found on its pre-existing urban fabric, such as regulations, transportation network, building morphology and economic factors. The GA parameters will be deducted from the recurring patterns observed in Tokyo's skyline growth, to later auto generate thousands of possible vertical growth scenarios over time, based on those patterns previously identified.

Fig. 84. Picture of Minato Ward skyline taken from Nishi-Shinjuku.(Photo by Author).

7.2 Tokyo's Historical Vertical Growth and Morphology: Minato Ward.

To develop and test the GA algorithm this thesis will focus on buildings over 130m tall exclusively in central Tokyo. The morphogenesis algorithm and methodology proposed will be directly applied to a 3D model. The first step is to identify and understand the recurring patterns that have generated the current Tokyo's skyline morphology, to later input the data into the base GA code. The next paragraphs will focus in the research and identification of those patterns through an exhaustive analysis of the historical data and morphology of Tokyo's high-rises.

By the end of 2011, a total of 116 buildings over 150m tall stood in central Tokyo, with more than 550 buildings more than 100m height located in the metropolitan area or larger Tokyo. In spite of a long-standing economic recession, Japan has actually experienced a huge boom in high-rise developments over the past 30 years, which has been particularly intense since the year 2000.

In order to understand Tokyo urban development, it is important to note that the city has been totally transformed three times during the twentieth century: first after the Great Kanto Earthquake of 1923, again after the Tokyo Air Raids of March 1945, and finally during the 1964 Olympics. The city has been totally rebuilt two times during the twentieth century, over the previous pre-existing urban fabric by keeping the ownership structure. During that period the city also multiplied its population by five, in particular during the postwar period, becoming the largest continuous metropolitan area in the world (Fig.85).

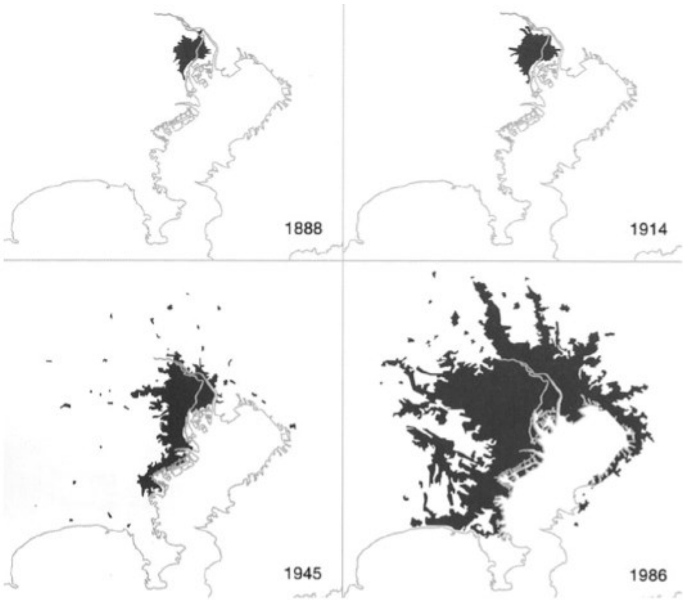


Fig. 85. Chronological expansion of Tokyo Metropolitan Area over the Kanto plain. (Source: http://masonmelnick.weebly.com/).

When the great Kanto earthquake hit the Kanto plain in 1923 the tallest building in Tokyo was the Ryounkaku building, a brick building 12 floors tall. The building had survived the previous 1894 earthquake, but this time collapsed on its 8th floor, and the construction of high-rise buildings was forbidden from that point on.

Is not until 1958 that Tokyo to gets its second high-rise structure, and first-ever super-tall Tower. The Tokyo Tower is a 332.5m emulation of the Eiffel Tower, just a few meters higher than its precedent. According to Seidensticker (2010) historian Umesao Tadao (1995) called the building a monument to Japanese lack of originality '…It seems that the main purpose of the tower was to rise a bit over the

Eiffel Tower and to lift up the spirits of Japanese people after the war.'

The first envisioning of Tokyo as a high-rise city happened also during that period, in the postwar years through the Metabolism movement. Part of the 'Metabolism' vision was based on mega-structures by overlaying large urban projects over the existing dense urban fabrics as an urban development strategy. Some 'Metabolist' architects proposed theoretical projects of towers to be superimposed over the existing city fabric, such as 'Tower City' by Kiyonori Kikutake in 1963 and the 'Octa-dwelling' by Kenji Ekuan in 1965, but these visionary projects were not further developed or built (Pernice 2004). While it is difficult to quantify how much influence the 'Metabolists' have had on modern-day Tokyo, their proposals had theoretical influence in later developments by transmitting the idea of imposing large-scale structures over an existing urban fabric. These early visions later became the validated trend in the development of high-rise structures in the city (Pazos 2014).

Regulations regarding earthquake resistance prohibited high-rise buildings until the 1960s, when office tower construction was permitted in certain areas of the city. In 1968 Tokyo's first truly high-rise office tower was completed the 36-story and 147m tall Kasumigaseki Building designed by Nihon Sekkei was the pioneer of modern office high-rise construction to the city.

Fig. 86. Minato Ward from in 1973, with the World Trade Center Building on the front right, and Tokyo Tower on the center left. Pazos (2014).

The Kasumigaseki Building was followed in the 1970's by a series of high-rise developments in Nishi-Shinjuku and Minato Ward with a few buildings with heights topping over 200m. These tower developments were located mostly on former public owned land, where regulations regarding high-rises were eased and master plans put in place. By the end of the 1970's the Shinjuku tower cluster was an isolated group of office towers built on the former water reservoir land; the Sunshine building stood alone a few kilometers north in Ikebukuro, while the World Trade Center and Tokyo Tower punctuated the sky above Minato Ward. The remainder of the city was indeed quite flat with very few structures of significant height (Fig. 86). There were few significant high-rise developments in the following decade, despite a strong economy. From 1980 to 1989, only four buildings over 150m were completed in central Tokyo, and all of them under 180m height. Land prices had skyrocketed in Tokyo during the 1980s, initiating the bubble economy. As the real estate bubble began to burst in the 1990s, a significant number of high-rise construction projects had already begun. Following the burst, land prices fell sharply and Japan fell into an economic recession (Krugman 2009). Ironically, an acceleration of high-rise construction happened during that time, despite the recession. Real estate values dropped considerably, urban regulations regarding high-rise construction were eased, and towards the end of the decade, the Japanese government injected capital into the economy. All of these factors fueled the development of high-rise buildings in Central Tokyo. By the beginning of the twenty-first century, the Urban Regeneration Act further eased regulations concerning Floor Area Ratio, thereby initiating a radical transformation process that led to the city's current state.

Fig. 87. Plan of central Tokyo showing the location of buildings over 150m tall completed by the end of 2011.

From the year 2000 to 2001, the city grew skywards with an incredible total of 73 buildings surpassing the 150m mark completed in central Tokyo. By the end of 2010, in addition to the Tokyo Tower and the Skytree, another 556 buildings with heights above 100m were standing in Tokyo Metro, with 106 of them taller than 150m in the center of the city (Fig. 87), making Tokyo one of the tallest cities in the world, ranking just behind Hong Kong and New York, CTBUH (2009).

Starting in 2009, Japanese Gross Domestic Product (GDP) fell sharply again when it seemed that it was starting to exit the long lasting economic slump, and construction in particular suffered a major slowdown. Yet, despite the recession the development of towers in Tokyo continued, and as of 2017 numerous high-rise projects are still under construction in the city, with more currently underway.

Fig. 88. Aerial Picture of central Tokyo, Minato Ward highlighted in red. (Photo: Hendrick Schicke).

One of Tokyo's most central administrative districts, Minato, will be used as a case study (Fig. 88). To develop the genetic algorithm, it is necessary to first precisely define the area of study as well as the number of high-rise buildings. To this end, the administrative district was selected so that similar conditions can be evenly applied to the whole area. It is also important to define the sample size or population of objects to be studied. Since the research study focuses on the morphology of the skyline, to run the algorithm, only buildings over 130m tall will be included. Thus, for the purposes of this article, a skyscraper or high-rise building will be defined as a building with an official height over 130m. According to the 'CTBUH Height Criteria' there is no absolute definition of what constitutes a "tall building", and different cutting marks are constantly used. What really defines a tall building is its height relative to its environment as well as the slenderness of the building. With this in mind, the model produced will focus on buildings that significantly stick out from its surroundings. Of the 124 buildings in Minato Ward over 100m, only 51 of them surpass the 130m mark, which is the threshold for a building to visually protrude out into the skyline in this ward (Fig. 89 to 91). Minato has been chosen as it contains 28.7% of the total buildings over 130m in Tokyo, including the two tallest buildings as well as the second tallest communications tower (Emporis Building Directory 2017). Oodaiba, a small area on the eastern side of Minato Ward, has been excluded from this study as it is an island in Tokyo Bay, disconnected from the main urban fabric and contains no buildings surpassing 130m height.

For the purpose of geometrically analyzing the morphology, location and number of buildings of the current skyline, a 3D photogrammetric model of Minato Ward has been used (Google Earth 2015). The model contains the terrain level as well as buildings over 15m, approximately four to five stories high, and accurately represents the Minato skyline in 2015 (Fig. 90).

To develop the genetic algorithms, economic data and parameters were taken from the existing urban fabric of Minato Ward. For this purpose, gradient probabilistic maps of the ward were developed from the 3D data. The gradient maps will inform regarding the probability of new high-rise developments to occur in a particular area.

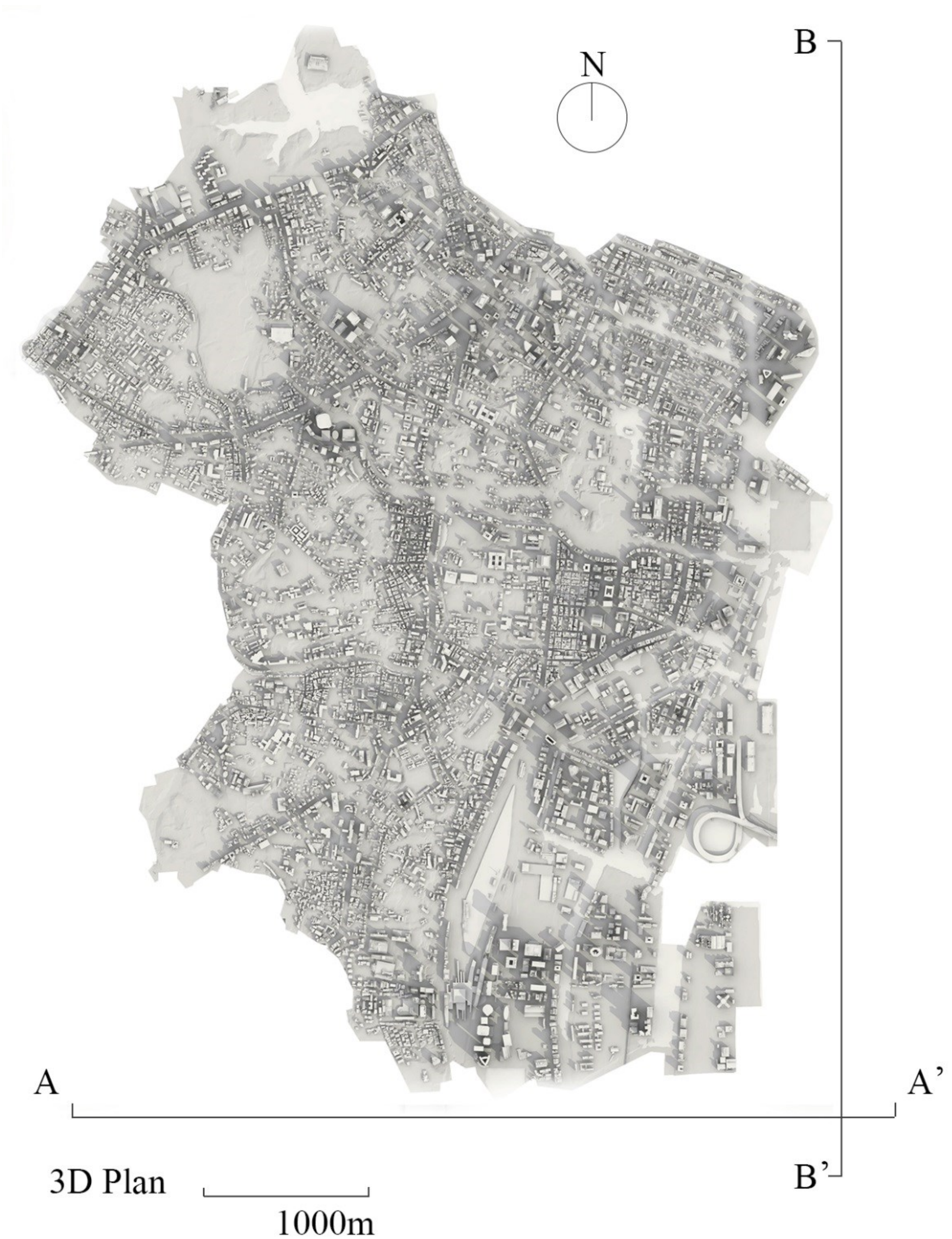


Fig. 89. Minato Ward. One of the most central among the Tokyo administrative 23 Wards. Plan view of 3D model of Minato Ward, showing buildings over 15 m height.

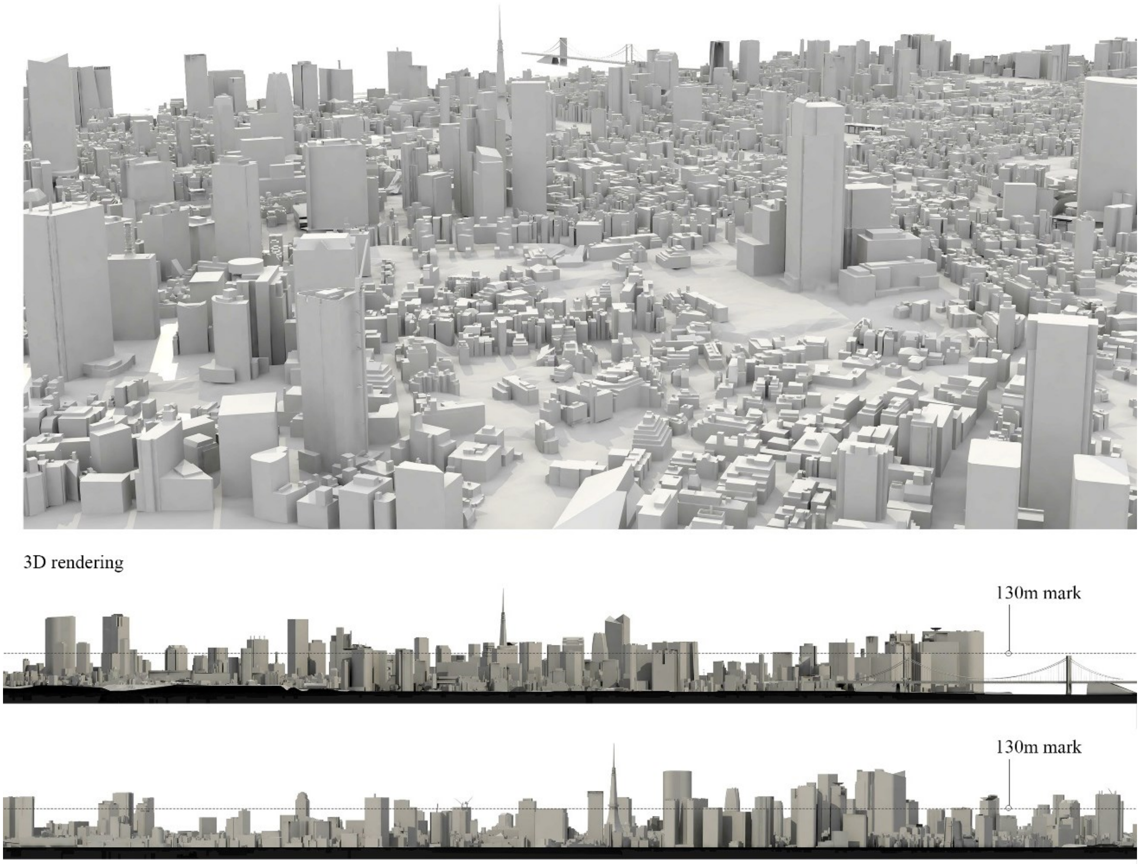


Fig. 90. VRay rendering of a polygon mesh photogrammetric 3D Model of Minato Ward skyline. Sectional-Elevation views of the Skyline, showing the 130m height mark.

For the initial research purposes, it will be assumed that the conditions that generated the current skyline over the past two decades will stay the same and that the skyline will continue to develop in a similar manner that it has been doing until now and since the start of the Bubble Economy in 1991. Once the parameters are set based in the current growth conditions, they could also be later modified to test other possible growth scenarios based on different patterns of growth, as the current conditions could change in the future. The algorithm could also work for different unforeseen regulatory or economic event, or even for catastrophic conditions, by imputing different parameters to adapt the equations to those unpredictable events, basically mutating the code (genotype) in order to adapt it to the changing conditions.

Fig. 91. Polygon mesh of photogrammetric 3D Model of Minato Ward skyline.

Fig. 92. Picture of the Minato Ward Skyline as 2016. (Photo by Author).

7.3 Urban Parameters: Land Ownership, Regulatory Master Plans, Vertical Urban Consolidation, Accessibility, Allocation and Economy.

To develop the evolutionary algorithm, the first step is to understand the simple rules and patterns that heave generated such a complex geometry. As Johnson (2001) has noted, self-organizing systems consist of very simple elements interacting with simple rules, from which complex behavior emerges. Pazos (2014) has identified the main factors that have determined the morphology of the Tokyo skyline: land ownership, regulatory master plans, vertical urban consolidation, accessibility, allocation and economy.

- Land Ownership: Public vs Private

The land ownership diagram (Fig. 93) shows publicly owned land in white and privately held land in black. The construction of private buildings is prohibited in public spaces such as streets, roads, parks, cemeteries, highways, train tracks, stations, etc. The municipal goverment has direct control over publicly owned areas which are designed and regulated as top down systems from a central local authority (Johnson 2001). In contrast, privately owned land is developed based on individual decision making as a bottom up process, although it must still adhere to building codes and urban regulations that set rules such as Floor Area Ratios (FAR), Building Coverage Ratios (BCR) and maximun permited heights. FAR regulations are variable in Minato Ward, with standard maximum percentages up to 1000% for residential and up to 1300% for high-rise office developments, and, even then, exceptions can be granted in excghange for the integration of public parks or facilities. The usual limiting factor in developing a building over 130m in height tends to be accumulating enough small land plots. Meaning that land reorganization processes are often neccesary, in order to put together land plots large enough to accommodate a high-rise building, based on the current FAR and BCR regulations.

- Land Reorganization Master Plans

Existing master plans already developed are shown in a consolidated diagram (Fig. 94), which shows the areas that have already undergone master plans through urban re-development processes in white. As Kogut (2006) has noted, the modern-day urban fabric of Tokyo is eerily similar to its origins in the Edo period (1603-1867) although the administrative boundaries have changed numerous times up until their current configuration which was decided in 1947. Tokyo high-rises have grown over this pre-existing urban fabric, with no substantial changes made to the previous land structure, except for ownership changes and localized regulatory master plans, involving land property reorganization. Almazan and Tsukamoto (2009) defined these types of high-rise developments as Corporate Urban Centers, which are an example of government policy reinforcing deregulation and privatization of urban developments. Meyer (2011) explains how this type of redevelopment strategy has now become the norm in Minato Ward, as finding plots large enough to build a high-rise building is very difficult due to the small size of land plots. Due to these reasons, areas compromising clusters of low-rise buildings are likely to be reorganized into privately driven high-rise redevelopments while areas that have been recently redeveloped are very unlikely to see new high-rise developments in the near future.

Fig. 93. Plan diagram of Minato Ward. Privately owned land is shown in black and publicly owned areas are shown in white.

Fig. 94. Consolidated Master-plans and public areas are shown in white. Non-planned privately owned land in grayscale.

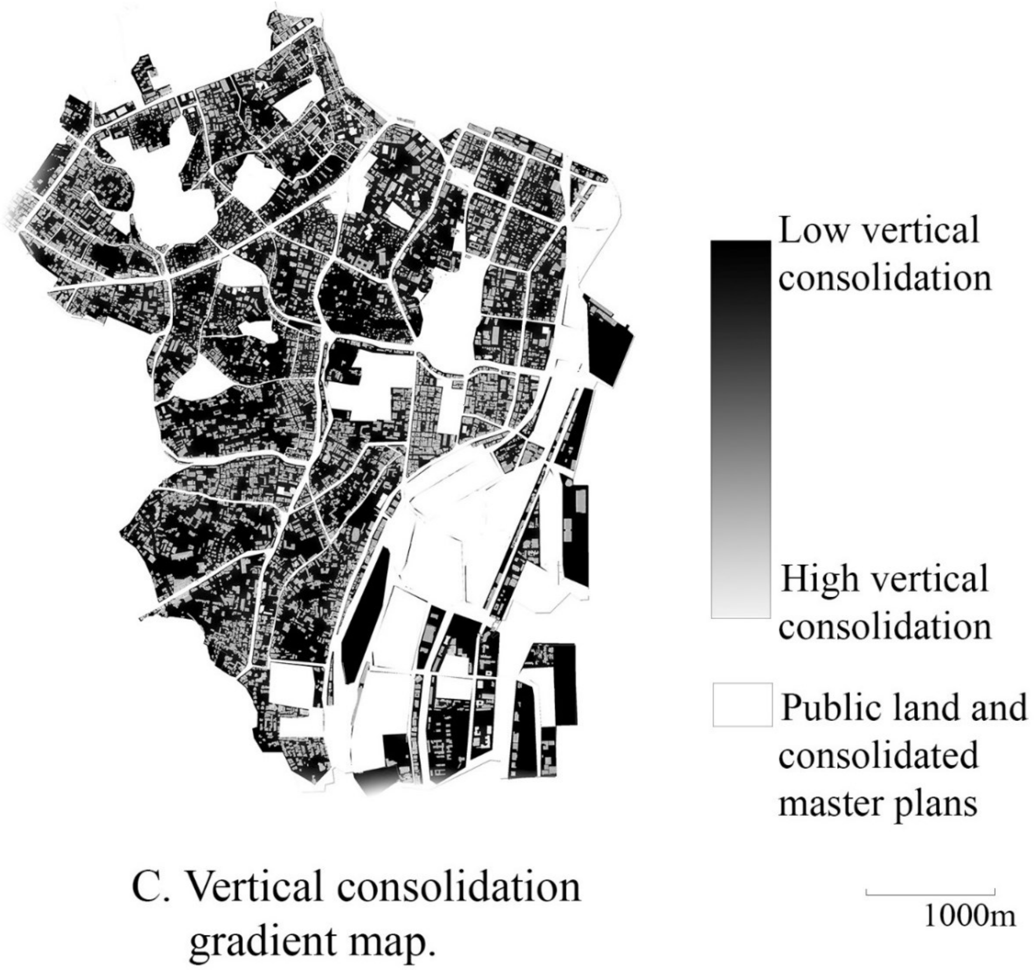


Fig. 95. Vertical Consolidation diagram. White areas show publicly owned land and consolidated master plans while the vertical consolidation is shown in a greyscale gradient.

- Vertical Density

The vertical consolidation diagram (Fig. 95) shows Minato Ward’s vertical density in grayscale. The darker shade represents lower-rise buildings which opens the possibility for new high-rise urban developments. This is due to the fact that Tokyo high-rise buildings are developed through the reorganization of small land plots which puts pressure on lesser dense areas to be developed into high-rises, as was explained in the previous paragraph.

- Accessibility

The train station proximity diagram (Fig. 96A) shows greyscale circles centered on subway stations with a radius of 500m (walking distance of 5 to 10 min). When several train lines overlap in one station, the circles overlap as well, producing a darker grey tone which indicates multiple access points to public transportation. In Tokyo, there is a strong correlation between the development of high-rise buildings and their proximity to public train transportation. When showing the footprint of buildings over 100m in black over the stations diagram this correlation of high-rise buildings and proximity to train stations is easily visualized (Fig. 96B). In the final accessibility diagram, the public space and already regulated spaces from the previous diagrams are subtracted (Fig. 96C). From this diagram, we can observe how the darker areas, which are in close proximity to public transportation, are more likely to experience further high-rise developments.

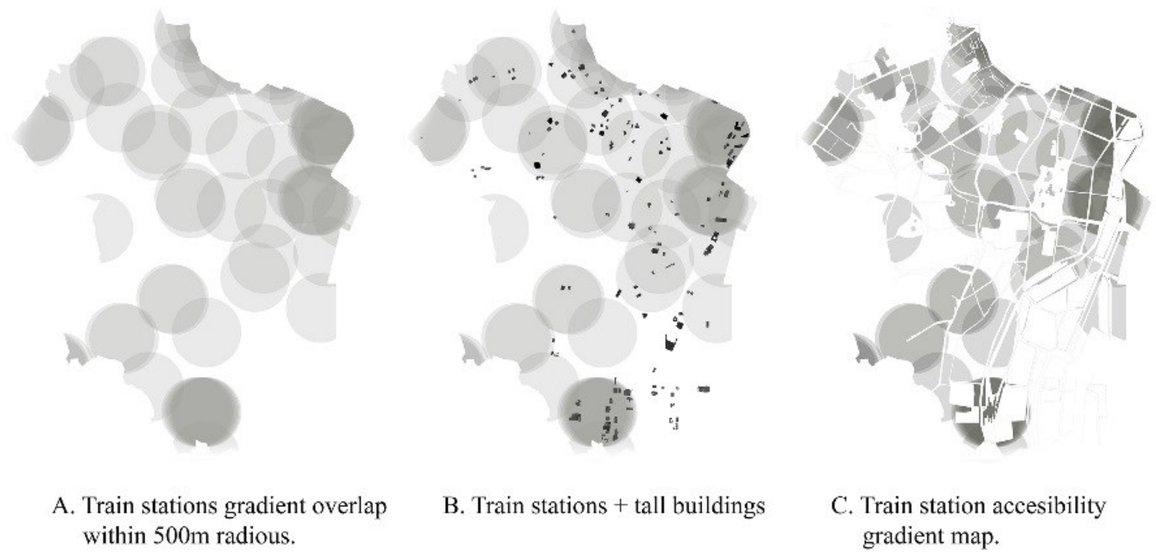


Fig. 96. Gray scale Diagrams of the subway stations with a radius of 500m (Walking distance of 5 to 10 min), and allocation of buildings over 130m.

- Allocation Parameters

With these various inputs, a final gradient probabilistic map combining the land ownership, existing master plans, vertical consolidation and accessibility was developed (Fig. 97), to be later used as the base for the computer model. This map is the result of the overlap of the vertical consolidation diagram and the accessibility diagram (Fig. 95 and Fig. 96C). The darker grey area reflects a high probability for future high-rise developments to occur due to less urban consolidation as well as its proximity to public transportation. Lighter grey areas represent a lower probability of further vertical growth and, finally, the white areas show where construction of vertical developments is prohibited or highly unlikely. The gradient plan was produced with simple parameters and will be used as a probabilistic base to determine areas with higher probability for new skyscraper construction.

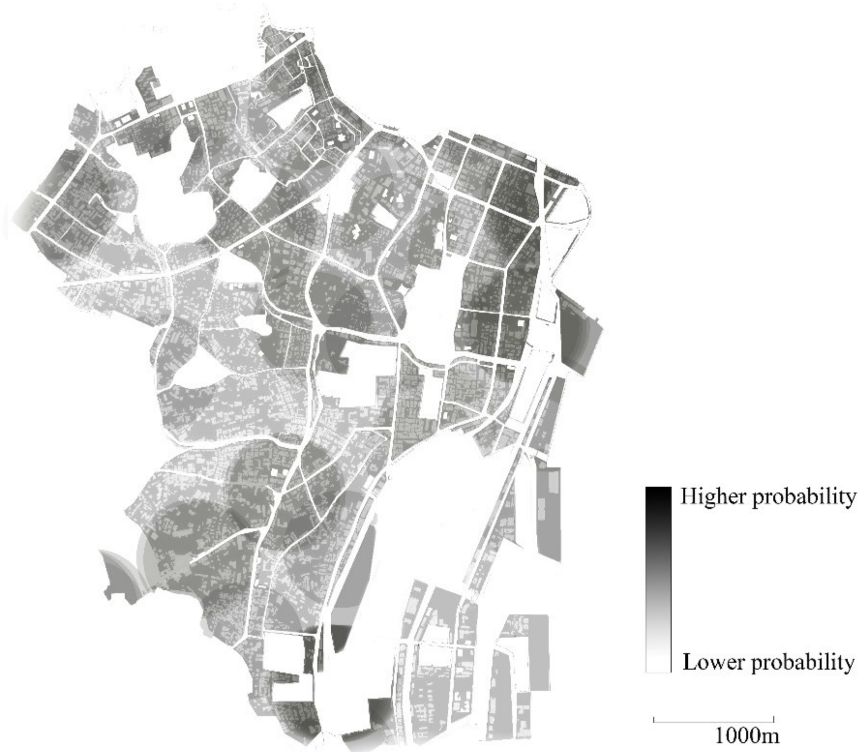


Fig. 97 Allocation diagram produced from overlapping Figs. 95 and 96C. The darker gray areas show the higher probability of future high-rise developments.

- Economic and Real Estate Parameters

The evolutionary process will incorporate economic and demographic data to predict the number of new high-rise developments annually. Understanding the overall economic context and its relation to the construction of high-rise buildings is an important factor. In 1991, Japan's economy entered a recession which, as Daniell (2008) has noted, was the result of a real estate bubble that was fed by easy access to loans using over-valuated properties as collateral. As Krugman (2008) explains, "Japan did not merely undergo a single year of catastrophic economic decline," rather the economy gradually slowed down with growth rates below 2% since 1992. Starting in the 1990s with the fall of land prices, some developers began acquiring significant numbers of small adjacent land in order to undertake large building projects in a way that had not been viable before. Most of the high-rise construction in Minato Ward followed this pattern of development. As Pazos (2014) has argued, high-rise developments in Tokyo tend to increase with low economic growth in part due to lower land prices, the introduction of economic stimulus, and less restrictive building regulations as the government attempts to boost productivity across the economy (Fig. 98 and 99). High-rise construction has been more of a tool to boost economic development than a result of economic growth itself. Since 1960, a total of 51 buildings over 130m were completed in Minato Ward with 31% of them during the year 2003 alone. The reason for this anomaly is that urban regulations regarding high-rise construction were eased in the year 2000 by the Urban Regeneration Act and the typical high-rise building takes an average of 3 years to build. According to Pazos (2014), attempts by the government to improve the economy have led to an increase in high-rise construction since each drastic fall of the economic growth rate results in more quantitative easing, which produces a wave-like pattern (Fig. 5). This economic patterns, and its relation to the number of buildings over 130m completed per year will serve as the base for the evolutionary computation process, in predicting of the number of future buildings per year.

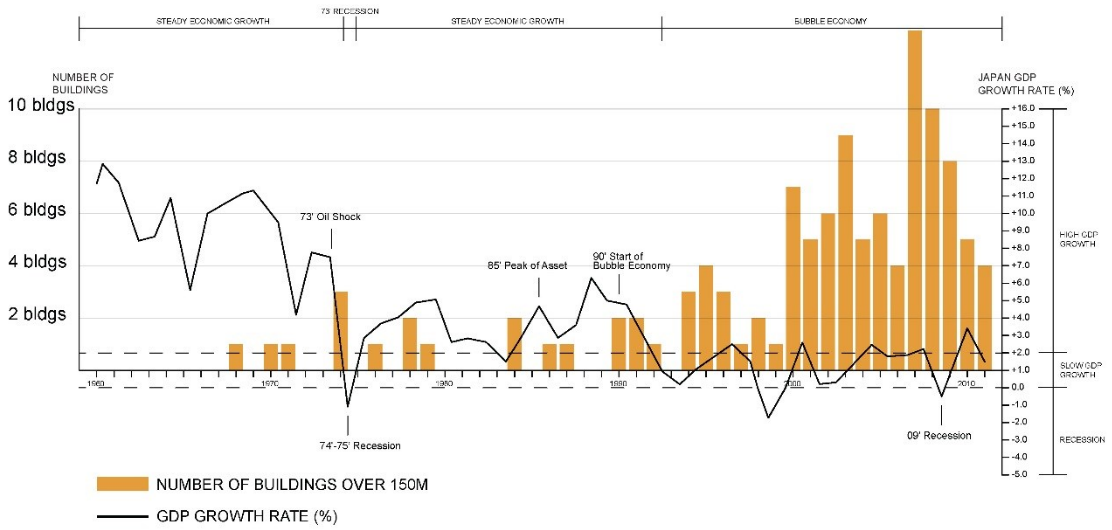


Fig. 98. Shows the number of buildings over 150 meters built in Tokyo since 1960 in bars, and the GDP growth overlapped on a graph.

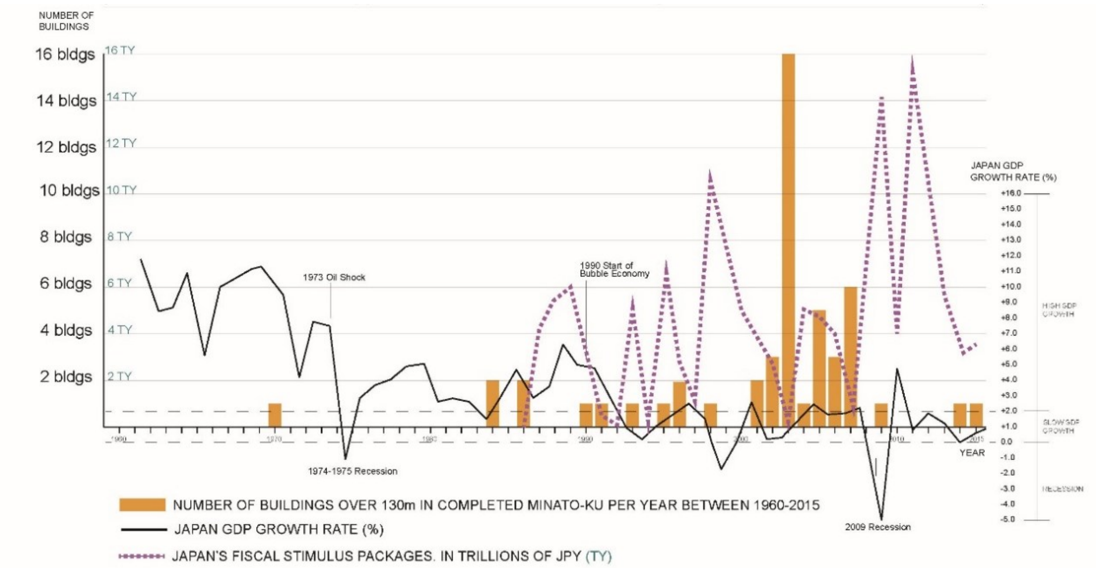


Fig. 99. Number of buildings over 130 meters built per year in Minato Ward from 1960 to 2015, combined with Japan's Gross Domestic Product (GDP) growth rates and economic stimulus packages.

7.4 Vertical Growth Evolutionary Algorithm.

By combining the probabilistic gradient map (Fig. 97) with the historical economic data (Fig. 99), a computer model to predict the construction of high-rise buildings over 130m in Tokyo's Minato Ward was developed and tested. This study used the gradient plan (Fig. 97) as a base to predict the most likely allocations for new high-rise buildings using a variety of determining factors: only areas where new developments are allowed, proximity to the public transportation network, the current location of high rise buildings, as well as variances in land prices, density, and population. The economic data, which was obtained from the World Bank (2016), covers the period from 1991-2015, when Tokyo experienced its high-rise boom. This data was then used to predict the number of buildings that would be built per year as well as the height of each building. The model is based on the assumption that the current conditions of vertical growth will remain constant and that there won't be major changes in the government's urban policy. In order to complete this task, the economic data for the region was used to statistically determine the number of buildings that would be developed within that area from 2016-2019 as well as their respective building height based on previous patterns of development and current urban regulations. The data parameters contain 184 economic indicators, such as population growth, fuel exports, foreign direct investment, deposit interest rates, etc. Regarding the construction data from 1991-2015, the data was obtained from specialized websites Emporis Building Directory (2017) and The Global Tall Building Database of the CTBUH (2017).

Once the economic data parameters were decided, a hybrid genetic algorithm was created, Mathias et al. (1994). The feature selection (Kudo and Sklansky 1998), feature transformation (Liu and Motoda 1998) and parameter selection (Hurvich and Tsai 1990) were made simultaneously to create an adjusted linear regression model using R-Squared as a measure of performance in the evolutionary process. The genetic algorithm uses mathematic operators to refactor input variables in order to find a suitable solution. All 12 available transformations in the evolutionary process used in this study are represented in diagrams (Fig. 100). Any continuous mathematical function could have been used; however, this specific subset was selected based on previous experiences.

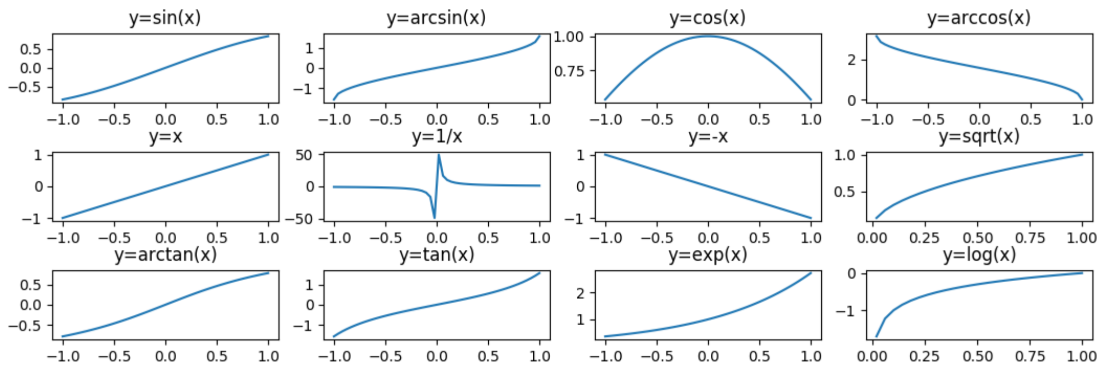


Fig. 100. The 12 available sets of mathematical transformations that were used to determine the evolutionary process.

The genotype of the individuals (high-rise buildings) is an array of the functions described above. The genetic algorithm attempts to calculate the best possible combination of selections and transformations for all the input features. The workflow of the hybrid genetic algorithm previously described is shown on a flowchart (Fig. 101). Through the evolutionary process, the best combinations of transformations in the input variables that maximize the previously determined objective function were determined.

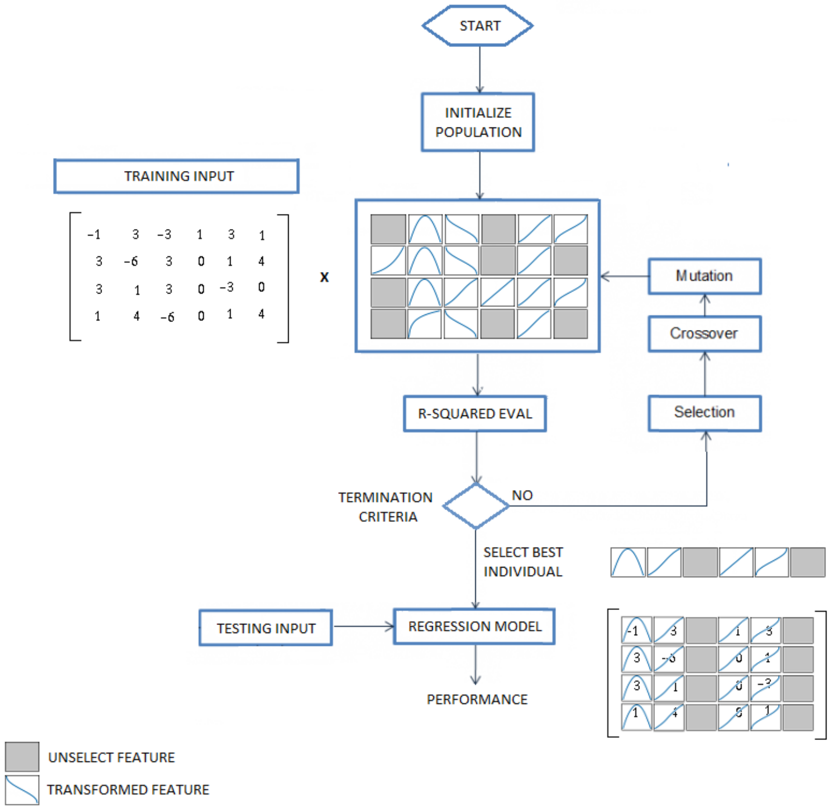


Fig. 101. Workflow diagram of the hybrid genetic algorithm. The Feature Selection, Feature Transformation and Parameter Selection were made simultaneously to maximize the objective correlation function (R-Squared).

During the feature selection process, the possibility of determining the usability of a variable when assigning null selection is known (Fig. 101). For the best individuals of the population, the parameters with the best adaptation were selected (parameter selector) to create the final regression model. The termination criteria of the evolutionary process are connected to the performance of the current individuals of a population in relation to the average individual of the population. When the average difference is lower than the preset threshold value for a homogeneous population, the iterative search process comes to an end. For the cases on which the threshold value is not reached a maximum number of iterations will be used to finalize the process.

Operators	Add, subtract, sin, cos, tan, asin, acos, atan, log, exp, sqrt, and inverse
Initialization	Ramped half-and-half
Fitness function	R-Squared correlation
Recombination Strategy	1-point crossover
Mutation Strategy	Leave-flipping
Mutation rate (pm)	0.05
Crossover rate	0.90
Selection Strategy	Proportional Roulette Wheel
Replacement Strategy	Invert-fitness

Table 04. Technical specifications of the algorithm proposed.

For this particular case, the genetic algorithm works over the entry data and attempts to maximize the R-Square regression value (Table 4). It uses a vector formed by the previously mentioned 184 indicators (variables) as training data and as objective data, the built buildings per year, as well as their median height.

Table 5 shows the capacity of the predictive model in relation to the determination coefficient R-squared and F-Test analysis (Seber and Lee 2012) for both the number of buildings and the median height respectively. All the results reported refer to the performance obtained in validation using 10-fold cross-validation and 50 independent runs. This probability is low enough to reject the null hypothesis using the common significance level of 0.05.

	Number of Buildings	Buildings Median Height
Best-Fit Values		
Slope	0.9900 ± 0.0229	0.9893 ± 0.0236
Y-intercept	0.0200 ± 0.0937	31.1800 ± 3.3660
X-intercept	-0.0202	-31.5200
1/Slope	1.0100	1.0110
95% Confidence Intervals		
Slope	0.9419 to 1.0380	0.9399 to 0.1039
Y-intercept	-0.1761 to 0.2161	24.1400 to 38.2300
X-intercept	-0.2240 to 0.1738	-40.2900 to -23.4600
Goodness of fit		
R-Squared	0.9899	0.9893
Sy.x	0.3731	9.8170
F-test significance		
F	1855	1755
DFn, DFd	1.1900	1.1900
P value	< 0.0001	< 0.0001

Table 05. Regression Analysis for both models

Equations 14 and 15 show the selected/transformed variables during the evolutionary search of the best adapted individual. For these individuals, the optimal parameters are calculated according to the input data to extrapolate the generated model.

$$
\begin{aligned}
\#\text{buildings} = {} & \tan(v0) + \tan(v1) + \sin(v2) + \tan(v3) + \text{asin}(v4) + \sin(v5) \\
& + \tan(v7) + (v10) + \tan(v11) + \sin(v12) + \text{asin}(v13) + \sin(v15) \\
& + \text{asin}(v17) + \sin(v18) + \tan(v20) + \tan(v21) + \text{asin}(v22) \\
& + \text{asin}(v23) + \sin(v25) + (v26) + \sin(v28) + \tan(v31) + \text{asin}(v32) \\
& + \tan(v35) + \text{asin}(v36) + \tan(v40) + \sin(v43) + \tan(v44) \\
& + \text{asin}(v45) + \sin(v46) + \text{asin}(v47) + \sin(v49) + \tan(v51) \\
& + \tan(v54) + \tan(v55) + \tan(v57) + \tan(v58) + \tan(v60) \\
& + \sin(v61) + \tan(v62) + \tan(v64) + \tan(v65) + \tan(v66) \\
& + \sin(v67) + \tan(v71) + \text{asin}(v73) + \tan(v74) + \text{asin}(v77) \\
& + \sin(v78) + \sin(v80) + \sin(v81) + \sin(v83) + \tan(v87) + \tan(v90) \\
& + \sin(v91) + \text{asin}(v92) + \tan(v93) + \sin(v94) + \sin(v96) \\
& + \sin(v97) + \sin(v99) + \tan(v100) + \sin(v102) + \text{asin}(v103)
\end{aligned}
$$

Equation 14. Variable selection / transformation formula obtained in the evolutionary search for Minato Ward used to predict the number of probable new buildings.

$$
\begin{aligned}
\text{avg}(\text{heights}) = {} & \sin(v1) + \text{asin}(v3) + \text{asin}(v7) + \text{asin}(v9) + \text{asin}(v10) + \sin(v13) \\
& + \text{asin}(v14) + \tan(v15) + \text{asin}(v16) + \tan(v17) + \tan(v20) \\
& + \tan(v22) + \tan(v25) + \text{asin}(v28) + \sin(v30) + \tan(v31) \\
& + \text{asin}(v32) + \sin(v33) + (v34) + \text{asin}(v35) + \text{asin}(v37) \\
& + \text{asin}(v38) + \tan(v39) + \text{asin}(v40) + \text{asin}(v42) + \sin(v43) \\
& + \sin(v44) + \text{asin}(v49) + \tan(v51) + \tan(v52) + \sin(v53) \\
& + \tan(v54) + \tan(v56) + \sin(v57) + \text{asin}(v60) + \tan(v61) \\
& + \tan(v62) + \sin(v64) + \tan(v68) + \tan(v69) + \text{asin}(v70) \\
& + \tan(v71) + \tan(v72) + \text{asin}(v73) + \sin(v76) + \tan(v77) \\
& + \sin(v79) + \tan(v80) + \text{asin}(v83) + \sin(v84) + \text{asin}(v85) \\
& + \sin(v86) + \tan(v87) + \tan(v89) + \sin(v90) + \text{asin}(v91) \\
& + \sin(v92) + \text{asin}(v94) + \text{asin}(v95) + \tan(v96) + \text{asin}(v99) \\
& + \text{asin}(v100) + \sin(v102) + \text{asin}(v105)
\end{aligned}
$$

Equation 15. Variable selection / transformation formula obtained in the evolutionary search for Minato Ward used to predict the average heights of probable new buildings.

Once both predictive models are determined, the gradient probabilistic plan is used (Fig. 97) for the generation of a stochastic roulette wheel based in Stochastic Universal Sampling (Baker 1987). A total of 100 independent simulations have been made according to the 2015 map by estimating the possible locations of the buildings in Minato Ward over the 2016 to 2019 interval. The number of buildings and its heights were determined using both predictive models. A probabilistic map was then generated where the darker greyscale tone represents higher likelihood of new building over 130m to be developed (Fig. 102).

	Observed real developments		GA Predicted developments		Difference (Observed-Predicted)	
Year	Number of buildings	Avg. height(m)	Number of buildings	Avg. height(m)	Number of buildings	Avg. height(m)
2016	1	230	0	0	+1	-230
2017	1	205	2	203.7	-1	1.3
2018	2	169.5	1	220.9	+1	-51.4
2019	2	202.5	3	278.8	-1	-76.3
Total	6	196.5	6	244.1	0	-47.6

Table 06. Real data for future construction in Minato Ward, versus predictions by the algorithm.

Data regarding future high-rise construction and the data predictions obtained from the computer model for the 2016 to 2019 period were compared to evaluate the results (Table 6). The left side of the table shows the real estimated data for future high-rise (undergoing construction), while the middle portion shows the data predictions by the computer model. The difference between both sets of data, are shown in the right column for evaluation purposes. According to the evolutionary model, a total of 6 new buildings over 130 meters should be built over this period. The table also includes the median height of the buildings and shows that a total of 6 buildings over 130m were undergoing construction or were planned for construction in Minato Ward from 2016-2019. The current developments under construction are: 'Sumitomo Roppongi Grand Tower' 2016 (230m), 'Akasaka Intercity Air' 2017 (205m), 'Park Court Akasaka' 2018 (170m), 'TGMM Shibaura' 2018 (169m), 'Toranomon Hills Residential Tower' 2019 (219m) and 'Toranomon

Hills Business Tower' 2019 (185m). It should be noted that there is the possibility that additional buildings over 130m that haven't been identified will be finalized before 2019. There is also a possibility that some of the buildings currently undergoing construction scheduled to be finished by 2019 could be delayed beyond the parameters of this case study. Due to these facts, it won't be possible to verify with total accuracy the observed real data until the end of 2019 and, thus, this data should be used only as an estimate for the purpose of pre-evaluating the results.

The algorithm simulation predicted 6 buildings in Minato Ward for 2016-2019 which matches with the at least 6 developments, either under construction or are planned to be completed, over the same period. Thus, the algorithm was 100% accurate in predicting the total number of buildings when contrasted with the observed current construction data over a four-year period. This is despite the fact that the algorithm had an error of one building per year, which suggests that, even if the algorithm was accurate in predicting the overall number of buildings, it was not accurate in predicting the exact time of construction as it deviates by a few months. Once again, it is important to note that construction delays or additional developments are still possible which might further skew these numbers. The algorithm prediction was not accurate regarding the average building height, with an average error of 47.60m or 19.50%. This result was probably due to the fact that the maximum possible building height entered into the algorithm was 300m. Even though buildings reaching 300m in height are allowed to be built in Tokyo (for example, there are several buildings of 300m which are planned to be completed in the 2020s), approval for such projects are only granted under special circumstances and currently there are no buildings over 260m completed in central Tokyo. Thus, placing a height limitation of 260m into the algorithm would have probably resulted in a smaller deviation. In order to test the algorithm's accuracy in predicting building heights, further research is necessary.

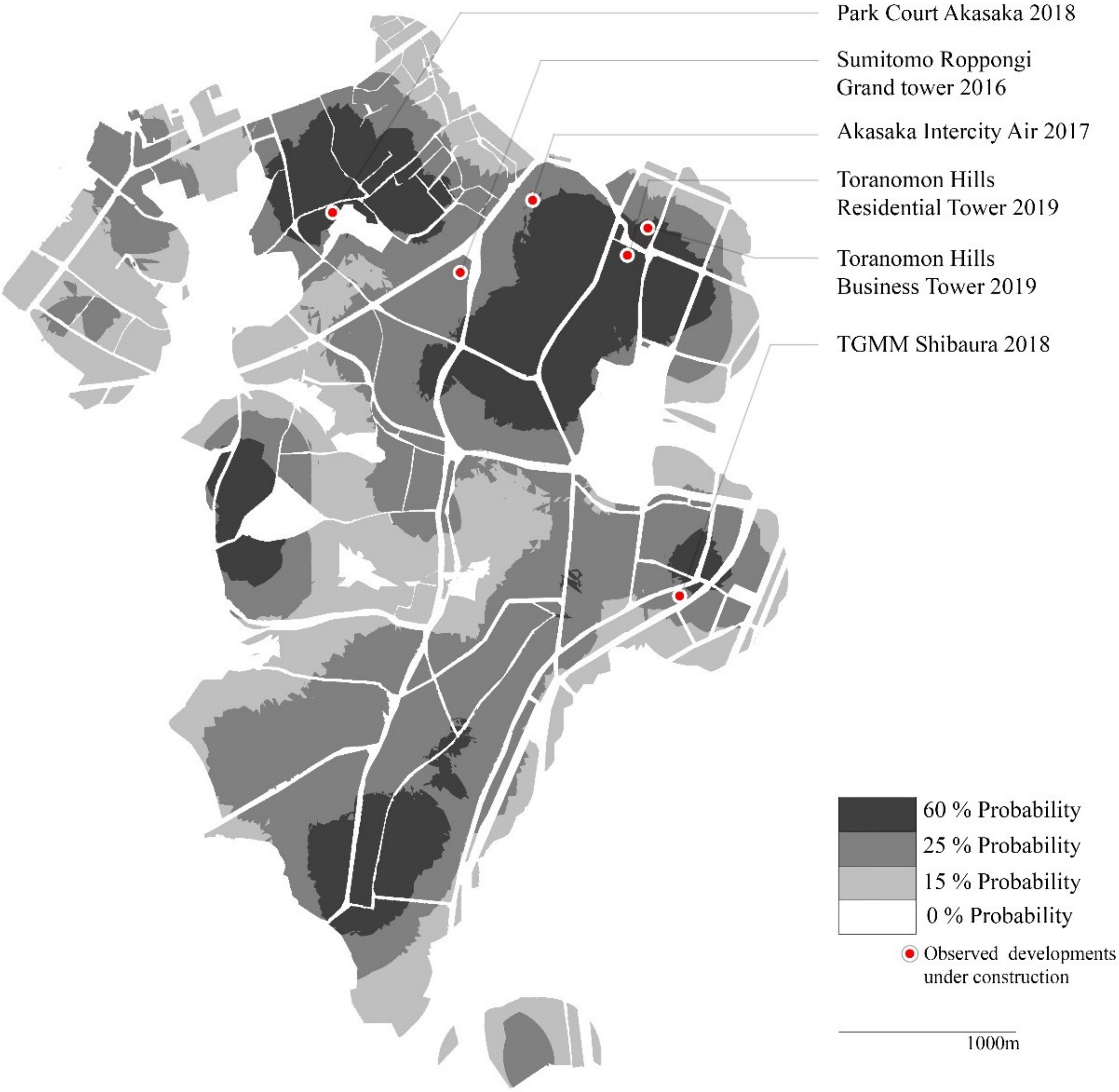


Fig. 102. Probabilistic gradient where new high-rise developments over 130m are likely to occur, darker grey shows higher probability. Highlighted in red are the high-rise buildings already planned to be completed on that area during the 2016-2019 interval.

The six dots on the grayscale map (Fig. 102) show the exact location of the already planned and undergoing construction projects to be built by 2019. A total of 4 buildings appear on the dark grey areas that the algorithm predicted and only two buildings, the 'TGMM Shibaura' and the 'Sumitomo Roppongi Grand tower', fall into the medium grey area, with one of them allocated right on the edge of an area of higher probability. It can be said that the results are in concordance with the probabilistic plan generated by the algorithm, with 66.67% of the buildings allocated in the dark gray zone versus a 60% result from the computer simulation, leaving a 6.67% margin of error. A 33.33% of the observed construction is allocated on the medium and light gray zones, versus a combined 40% (25%+15%) by the computer model, again with a margin of error of 6.67%. Due to the small size of the population sample (6 buildings), these deviations are considered acceptable.

7.5 Morphological Parameters: Plan Type, Area and Heights.

In addition to the location and average height of the new high-rise developments, its quantity, size and massing geometry are of interest to this doctoral thesis, as they will define the overall morphology of the skyline. What will determine the final geometry and look of the skyline is the overall number of buildings, their exact location within the urban fabric and its massing morphology. A classification of the current existing buildings by its volume and typology reveals recurring formal patterns that will inform the morphology of future high-rise developments (Fig. 103).

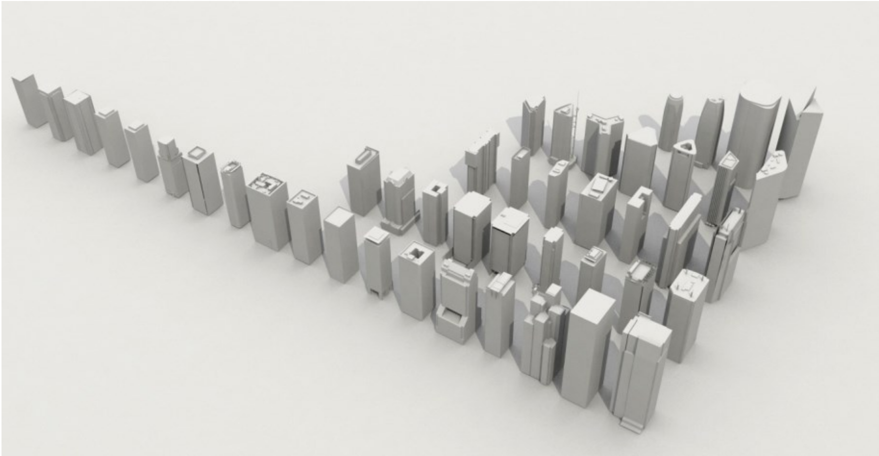


Fig. 103. 3D model classification of the 45 Minato Ward's buildings over 130m by building height and plan footprint geometry.

The existing buildings in Minato Ward have been classified by its plan geometry, building footprint, area, overall height and slenderness (Fig. 104). High-rise constructions have characteristic recurring geometries and share common morphological characteristics, as most building typologies have very specific requirements based in floor plan efficiency, function, natural light, structural integrity, egress, vertical circulation, construction methods and economic efficiency.

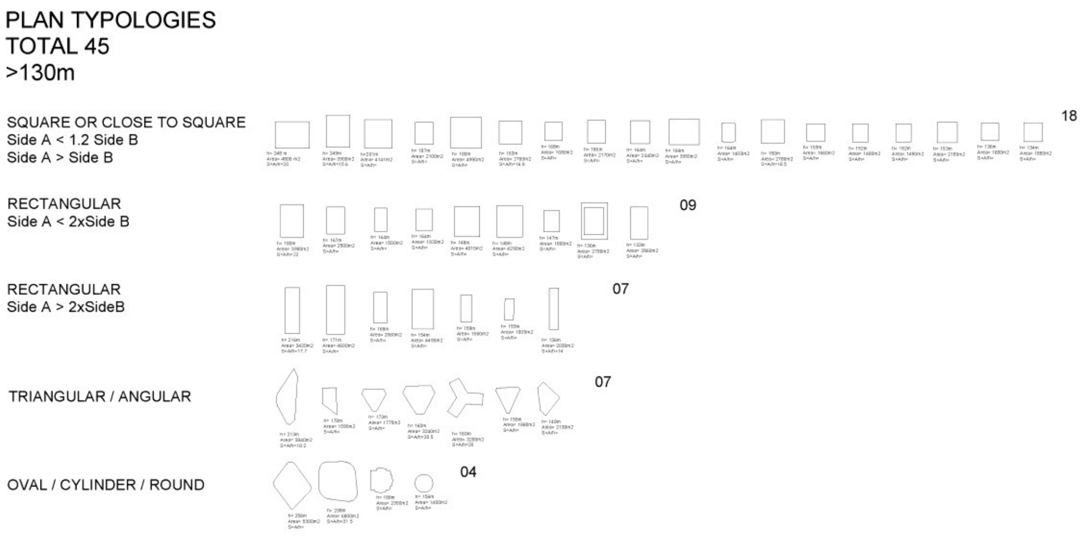


Fig. 104. Classification diagrams of Minato Ward's buildings over 130m by its plan footprint geometrical typology, showing recurring geometrical patterns.

As shown on Table 07, a total of 75.6% of the buildings have rectangular or square plan geometry, as it is the by far the most efficient and economic geometry. Also but in less extend are found some other geometrical configurations such as polygonal, triangular, cylindrical or oval geometry.

Plan Typology	Number of buildings	Percentages
Square (Almost Square)	18	40.0%
Rectangular A<2B	9	20.0%
Rectangular A>2B	7	15.6%
Triangular / Angular	7	15.6%
Round / Oval	4	8.9%
TOTAL	45	100%

Table 07. Percentage of buildings over 130m in Minato Ward, classified by its plan footprint geometry.

Table 08, shows how a total of 84.4% of the buildings are in between 130 to 200m, with only 15.6% over the 200m mark, with and overall height average for buildings over 130m of 170m.

Building Heights	Percentages
250 to 280 m	2.2% (1 building)
220 to 249 m	6.7% (3 buildings)
190 to 219 m	6.7% (3 buildings)
160 to 189 m	42.2% (19 buildings)
130 to 159 m	42.2% (19 buildings)
Average height:	170.02m.

Table 08. Building height classification by percentages.

The footprints areas vary between 1000m2 to 7000m2, being the most common area in between 1500m2 to 3000m2 with a total of 53.3% of the building within that range, as it optimizes structure, vertical circulation, usable area and natural light. The average building footprint area is 2,788m2, as shown in table 09.

Building Areas	Percentages
1000m2 to 2000m2	35.6% of buildings (16)
2000m2 to 3000m2	28.9% of buildings (13)
3000m2 to 4000m2	15.6% of buildings (7)
4000m2 to 5000m2	15.6% of buildings (7)
5000m2 to 6000m2	2.2% of buildings (1)
6000m2 to 7000m2	2.2% of buildings (1)
Average area:	2788 m2

Table 09. Towers footprint area classification by percentages.

As the relation between floor plan area and building height tend to be proportional, taller buildings in general will have larger plan footprints in order to maximize the vertical circulation and usable floor area. Another factor is structural stability, as very slender tall buildings will not have a good structural behavior to lateral forces such as seismic and wind stresses. Slenderness for the purpose of this study has been defined as the footprint area divided by its total height:

$$S = {}^{A}/_{h}$$

Equation 16. Slenderness (S)

A total of 80% of Minato's buildings over 130m have a slenderness ratio between 8.5 to 22, with an average of 16.2, table 10.

Building Slenderness	Percentages
A/h= 5 to 10	17.8% of buildings
A/h= 10 to 15	35.6% of buildings
A/h= 15 to 20	17.8% of buildings
A/h= 20 to 25	15.6% of buildings
A/h= 25 to 30	13.3% of buildings
Average Slenderness (A/h)	16.2

Table 10. Towers slenderness classification by percentages.

	Tower Area	Height	Slenderness
Averages	2788m2	170m	16.2

Table 11. Show the Overall averages of existing buildings over 130m in Minato Ward.

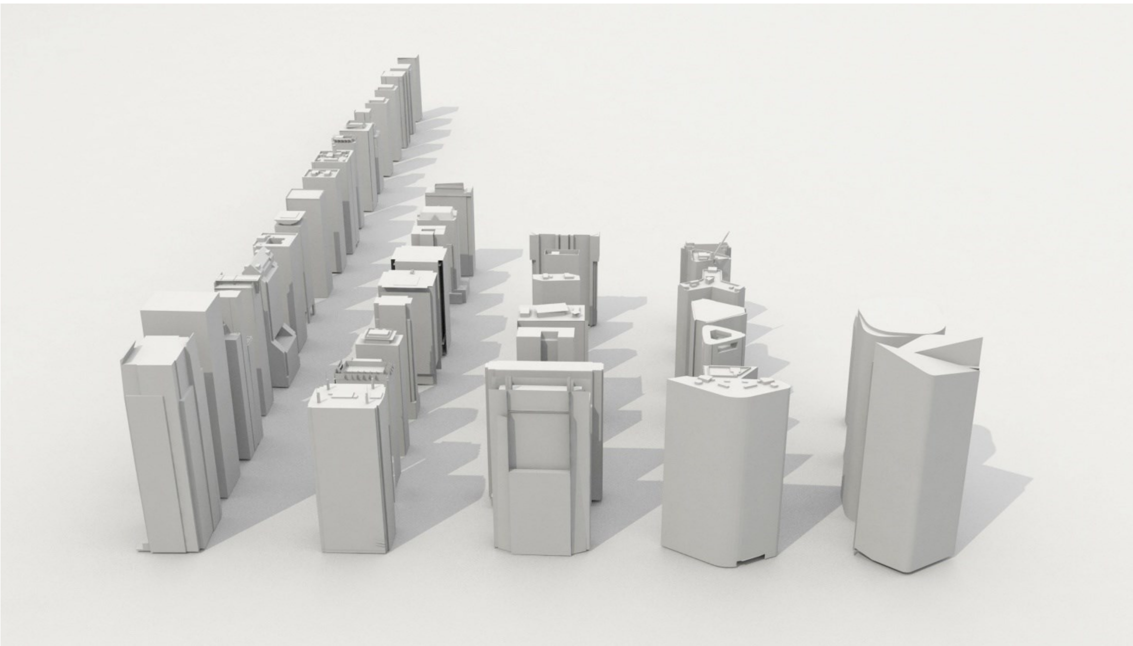


Fig. 105. 3D diagram showing a close-up of the different building morphologies of Minato Ward buildings over 130m.

Fig. 106. Picture of Minato Ward skyline,2016. (Photo by Author).

7.6 Skyline Parametric Morphogenesis.

The previous sub-chapter analyzed the massing geometry of the existing high-rise buildings over 130m in Minato Ward, identifying the recurring morphological patterns of Minato Ward's high-rise buildings. Those parameters were used in combination with an evolutionary algorithm, to simulate the evolution of the morphology of the skyline. Table 12, shows the results by the proposed GA.

	GA Predicted developments	
Year	Number of buildings	Avg. height(m)
2016	0	0
2017	2	203.7
2018	1	220.9
2019	3	278.8
TOTAL	6	244.1

Table 12. Predictions of new high-rise developments until 2019 by the GA algorithm.

The proposed methodology is a combination evolutionary computation and parametric process. Self-organizing systems as cities, show recurrent patterns on its growth, but it is important to note that randomness encounters or interactions play a crucial role on those systems (Johnson 2001). In that sense the possibilities of combination of the different parameters of a self-organized system are infinite. Some combinations however are much more likely to occur than others. All of the solutions will be different but will share similar characteristics. The process is similar to a biological natural system, as all the individuals of the same species will have similar characteristics, and will share a common biological structure, but none of them will be identical. In that sense the GA and parametric methodology proposed in this thesis will not predict the future of Minato Ward high-rise developments, but will auto-generate very likely solutions, somehow generating solutions that will have a similar morphology and distribution that what most likely ones to happen in the future. The proposed methodology will generate random scenarios with high probability to occur, particularly from a morphological point of view.

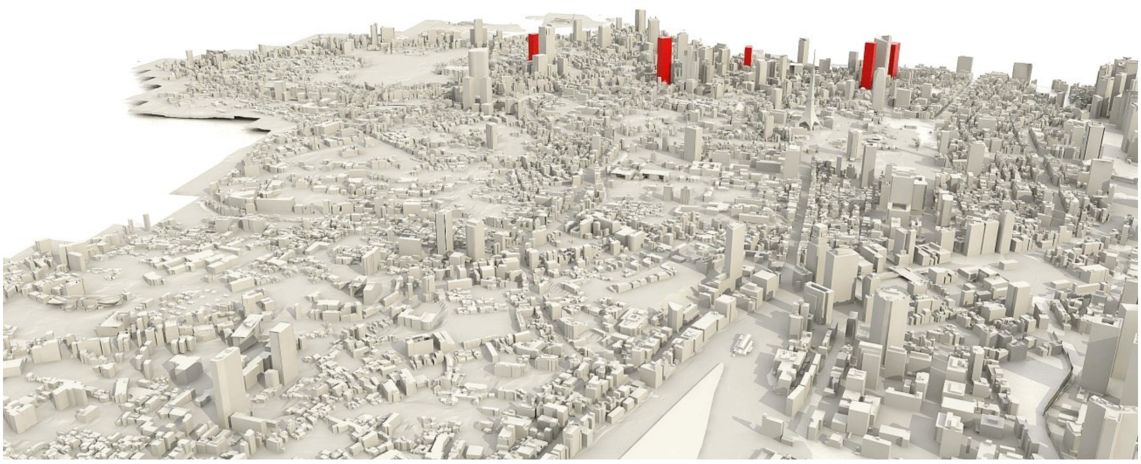


Fig. 107. Buildings planned (or under construction) to be completed between 2016 and 2019 are shown in red, over the current skyline.

Random individual decisions impossible to predict made by land owners, banks, developers, politicians, citizens, architects, planners, etc. will play a key role in the city transformation for each independent location. Also, individual events such as inheritances, business deals, technological developments, catastrophic events, investments, loans, etc. will have an impact on whether a new high-rise development will occur on a specific location. The proposed methodology will use a random stochastic roulette (Baker 1987) methodology instead, within pre-stablished parameters, equivalent to the randomness of a self-organizing system. The thesis will develop one example of growth as an example, but as previously mentioned the possibilities are literary infinite as in any biological system.

Building shapes, heights, areas and slenderness will be based on the parametric relations shown on tables 13 and 14. By random generation of values within those pre-stablished parameters, the values on tables 108 and 109 have been generated, they show just two random set of values, that meet the required parameters. The data has been generated by random assignation of values within the pre-established parameters. The data obtained are just two possible examples within the infinite possible combinations.

Plan Typology	Tower area	Height	Slenderness
Square Plan	2232m2	156m	14.3
Square Plan	3673m2	138m	26.6
Rectangular A<2B	1283m2	178m	7.2
Rectangular A<2B	1758m2	157m	11.2
Triangular / Angular	4329m2	192.4m	22.5
Round / Oval	2957m2	160m	18.9
Averages	2704m2	163.5m	16.7

Table 13. Example A. Shows an of 3D data, based on the identified parameters.

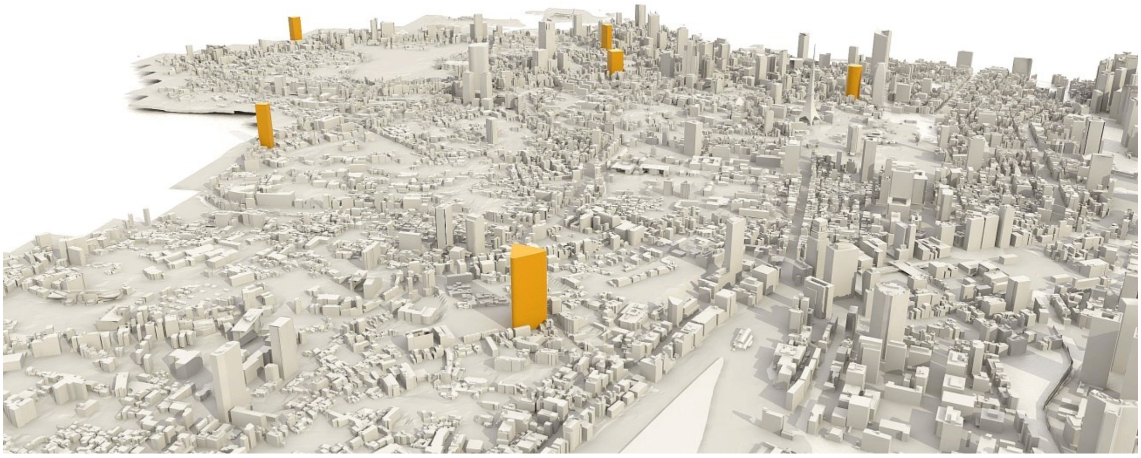


Fig. 108. Example A. In orange, the 3D simulation of the results obtained for example A.

Plan Typology	Tower area	Height	Slenderness
Square Plan	3455m2	143.3m	24.1
Square Plan	1789m2	130.5m	13.7
Rectangular A<2B	1521m2	158.4m	9.6
Rectangular A<2B	4732m2	167.8m	28.2
Triangular / Angular	2777m2	241.5m	11.5
Round / Oval	2367m2	142.6m	16.6
Averages	2773.5m2	164m	17.2

Table 14. Example B. Shows an of 3D data, based on the identified parameters.

Fig. 109. Example B. In blue, the 3D simulation of the results obtained for example B.

The data resulting from the parametric random process, has generated the two previous examples, which have been mapped over the areas of high probability generated by the GA, on sub-chapter 7.5. The areas in dark gray have been assigned to a 60% probability and the areas in medium grey a 25% probability and the areas in lighter grey shade a 15% probability. The results have been modeled in 3D and overlapped over the current 3D model of the Minato Ward skyline. The two resulting skyline interactions generated by a combination of an evolutionary and parametric process are shown on Fig. 108 and 109, for examples A and B respectively.

When the results obtained are compared with the buildings currently under construction, planned to be completed before 2019 (Fig. 107), the results are different, however of similar density, distribution and geometry.
The two examples shown in orange and blue (Fig. 108 and 109) are just two random samples of the infinite possible combinations generated by the evolutionary and parametric process proposed. In that sense the proposed methodology do not exactly predicts urban vertical growth, but simulates real city growth, bringing the boundaries between real artificial closer together. On Fig. 110 the overlap of the real high-rise buildings under construction in red, and the examples A and B generated by the proposed methodology, are shown all together.

Fig. 110. Shows in Red real buildings planned to be completed between 2016-2019. In orange and blue two different examples of buildings generated by the GA and parametric algorithms.

7.7 Results.

In the same manner as organisms (Johnson 2001), cities experience constant change and transformation through endless mutations in what constitutes the ultimate and most visible expression of civilization. The constantly changing skylines of large urban centers have come to define their identity. The morphological evolution of cities and biological growth are both driven by a self-organizing process. In this case study, an adaptive evolutionary model was tested through the use of genetic algorithms to predict the likeness of future vertical growth in Tokyo's Minato Ward. First, the areas with high potential for future high-rise developments based on previously identified recurring patterns of growth (regulations, vertical density and accessibility) were identified over a gradient map. Then, using data from previous economic patterns and high-rise construction, the algorithm predicted the number of new buildings expected to be built per year and their respective heights to generate a probabilistic map of new buildings.

The results obtained from the proposed approach were then compared to the real construction data of buildings over 130m planned to be completed in Minato Ward by 2019. After testing the genetic algorithm predictions for the 2016 to 2019 period and contrasting the results with the real projects underway, it can be concluded that the growth estimates by the algorithm were accurate regarding the total number of buildings (100%) and their likely locations (+/-6.67%). However, the algorithm didn't accurately predict exact year of the developments (+/- one year) and the height of the buildings (19.50% deviation), suggesting that further studies should be done on those areas. In future case studies using this methodology, it should be proposed that a larger sample area be tested since this current study was partially limited by the small size of the population sample (6 buildings). Nonetheless, it can be concluded that the use of evolutionary computation yielded acceptable results when used to predict future urban vertical growth.

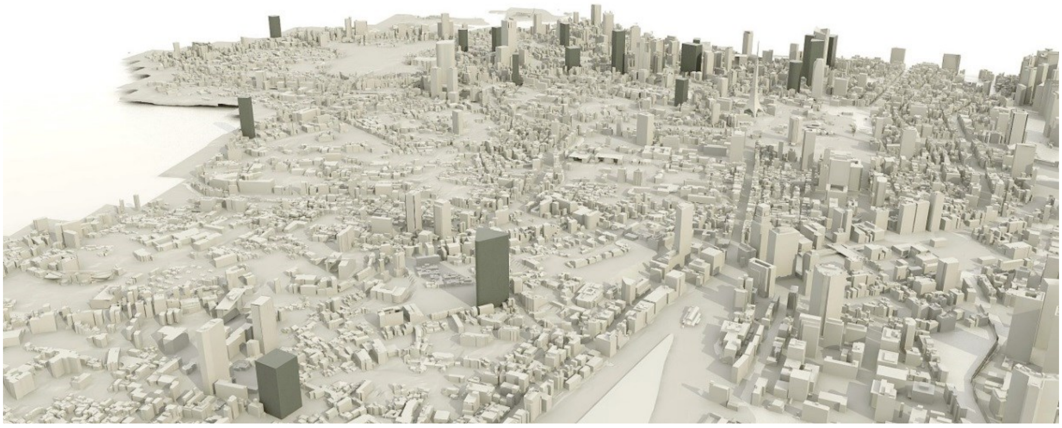


Fig. 111. Shows in darker grey future high-rise developments, as one example out of the infinite possible scenarios, of a 12-year evolutionary growth of the Minato Ward skyline.

In Addition, based on the identified recurring patterns, a parametric process in combination with the biological evolutionary algorithm has simulated two different examples of vertical growth.

The algorithm estimates first different patterns of growth, or buildings to be built per year. Then allocated those buildings on Minato Ward plan, based on the likeness for new developments over Fig. 102 diagram, to then generate the building mass morphologies, which were randomly determined by random combination of the pre-stablished parametric relationships.

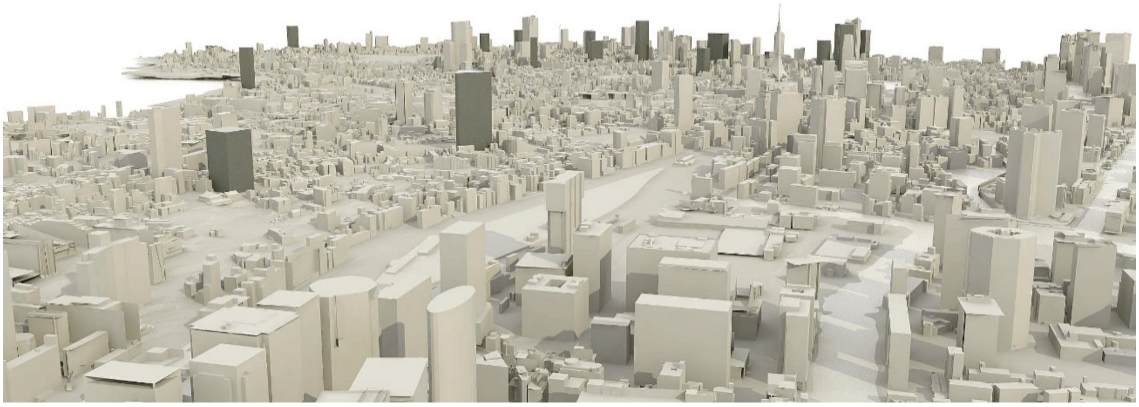


Fig. 112. Shows in darker grey future high-rise developments, as one example out of the infinite possible scenarios, of a 12-year evolutionary growth of the Minato Ward skyline.

Figs. 111 and 112, show a hypothetical example for a 12-year period (from 2015), as one of the many possible scenarios of how the Minato Ward skyline could look by 2027. The proposed methodology has been able to generate real evolution of a complex system based in pre-identified patterns of behavior, creating real 3D objects before they happened, furthering blurring the boundaries between 'real urban growth' and 'artificially generated urban growth' (Fig.112 & 113).

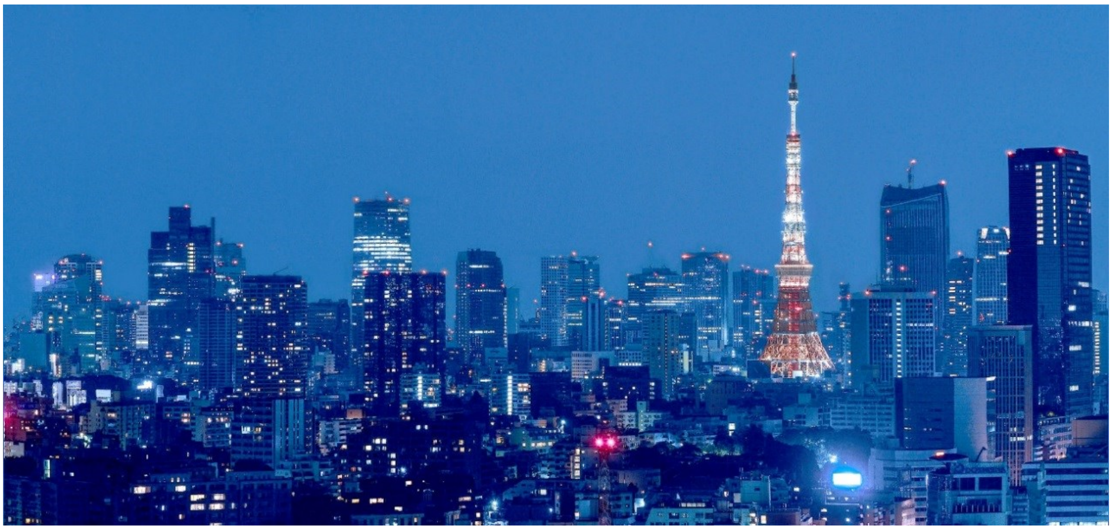


Fig. 113. Partial aerial view of the Minato Ward skyline, as 2016 (Photo: Inefekt).

8 CONCLUSIONS

After the evaluation of each case study results, individual and overall conclusions are drawn. The main hypothesis stated that by introducing recently developed technologies and advanced computation techniques, a greater level of integration between 3D models obtained directly from the 'real world' and 'artificially generated' 3D models (that do not exist in the 'real world') could be achieved, further blurring the boundaries between 'real' and 'artificially generated' 3D models. To test the hypothesis an algorithm has been proposed each one of the sub-hypotheses:

8.1 Sub-Hypothesis 1.

An algorithm was proposed for the conversion of NURBS and polygon meshes into point clouds; it was applied and tested in Chapter 5 into a model of a croissant, a house and a historical plaza. The case study proposed a PBR methodology and a converting algorithm to export polygon meshes and NURBS to point cloud format. Realistic renderings were produced by combining objects from the 'real world' with 'artificially generated' objects. The quality of the results was photorealistic. It is also possible to achieve similar quality with other commonly used modelling software's and rendering engines. However, the results obtained by using the proposed algorithms were more accurate and precise, as they were obtained directly from exact and precise laser measurements of the 'real world'. Future developments in PBR will make this type of rendering techniques more efficient, and most likely will become of a common representational tool in architecture and urban design. The proposed methodology conveys a great degree of accuracy, realism and time savings, yielding excellent rendering results.

8.2 Sub-Hypothesis 2.

A genetic Algorithm was proposed to generate through evolutionary computation diversity and deformations on a croissant's surface, a tile and a skyscraper's facade design. Chapter 6 proposed the use of a IEC algorithm, to generate diversity and irregularity through a morphogenesis process. Satisfactory results were achieved on the 3D modelling of complex geometries. The evolutionary computation algorithm could generate diversity from an original sample automatically, by using a natural selection process, within pre-stablished parameters. The proposed methodology had also proven successful for the auto generation of designs, when applied to a skyscraper façade design.

8.3 Sub-Hypothesis 3.

A Genetic Growth Algorithm was proposed and then applied to a real city's model to generate possible scenarios of future vertical growth. On Chapter 7, an adaptive evolutionary model was tested through the use of genetic algorithms to predict the likeness of future vertical growth in Tokyo's Minato Ward. First, the areas with high potential for future high-rise developments based on previously identified recurring patterns of growth (regulations, vertical density and accessibility) were identified over a gradient map. Then, using data from previous economic patterns and high-rise construction, the algorithm predicted the number of new buildings expected to be built per year and their respective heights to generate a probabilistic map of new buildings.

The results obtained from the proposed approach were then compared to the real construction data of buildings over 130m planned to be completed in Minato Ward by 2019. After testing the genetic algorithm predictions for the 2016 to 2019 period and contrasting the results with the real projects underway, it can be concluded that the growth estimates by the algorithm were accurate regarding the total number of buildings (100%) and their likely locations (+/-6.67%). However, the algorithm didn't accurately predict exact year of the developments (+/- one year) and the height of the buildings (19.50% deviation), suggesting that further studies should be done on those areas. A parametric probabilistic random process was then proposed to successfully model possible scenarios of urban vertical growth. The final output was an 3D model resulting from an evolutionary process of an 'artificially generated' skyline. The 3D model simulation accurately reflects the 'real' vertical growth of the city.

By simulating possible scenarios of vertical urban growth, urban planners, policy makers and designers will be able to better assess future changes in cities and anticipate the necessary responses for implementing new infrastructure or regulations. Through imputing different economic, legal, regulatory and real estate parameters, planners will also be able to assess what will happen if those conditions change.

8.4 Final Conclusions

Based on the evaluation of the results it can be concluded that the use of the proposed computation techniques and evolutionary algorithms are an efficient and promising way to achieve more integration in between 3D data from the 'real world' and 'artificially generated' 3D data.

The use of LIDAR and PBR techniques yielded more accurate rendering results than other more commonly used modelling and rendering techniques.

The artificial intelligence algorithms have proven to be successful into achieving more realism by artificially generating superficial irregularities and diversity through evolutionary computation.

The use of evolutionary computation algorithms has also proven successful in predicting the growth and evolution of complex self-organized urban environments.

Overall it can be concluded that the proposed use of point-based techniques and artificial intelligence processes has successfully achieved more integration between 3D models obtained directly from the 'real world' and 'artificially generated' 3D models, both visually and geometrically, as it was postulated in the main thesis hypothesis (Fig. 114).

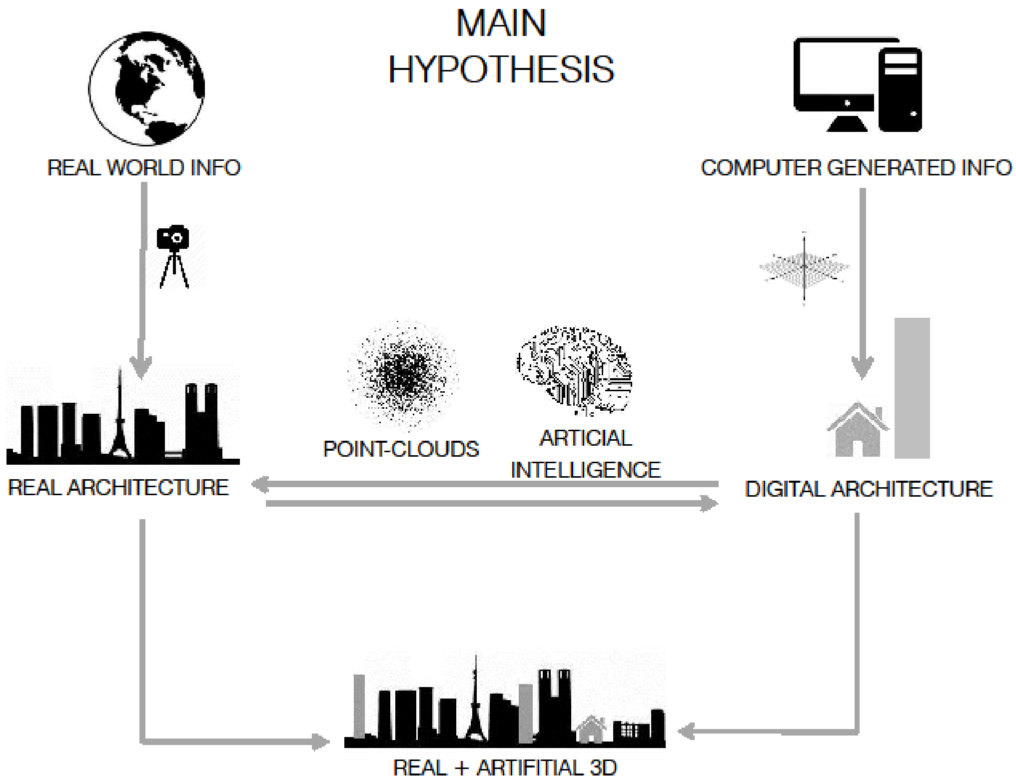


Fig. 114. Main thesis hypothesis.

9 FUTURE DEVELOPMENTS

All the mentioned process are still novel techniques not commonly used in the architecture industry due to its complexity, as most of them are still under development and not widely commercialized. Three research areas to be further developed in future studies have been identified:

9.1 PBR and Point Clouds.

As mentioned in previous chapters PBR is a relatively new rendering technique not commonly used in Architectural rendering. Further developments in PBR rendering will make this efficient rendering technique more common. Commercialization of PBR software packages will most likely result in a higher investments and further developments on this promising rendering technique. Further developments in the translations of models from NURBS and mesh formats into point cloud formats, and vice-versa will bring advances in the practical use of LIDAR technologies.

9.2 Evolutionary Morphogenesis.

This novel technique is not commonly used due to its complexity. The incorporation of the proposed algorithms into commercial software packages could lead to a methodological reformulation of the design discipline, not just but facilitating the work, but by transforming how computers work from just a production machine into intelligent design machines. This methodology could aid designers to find the most appropriate solutions to a problem. The introduction of evolutionary computation and machine learning into the design process, could eventually lead to a re-formulation of the design methodology of complex geometries.

9.3 Self-Organizing Growth.

Incorporating evolutionary growth algorithms, able to learn from the rules and parameters of self-organizing systems could suppose a break through development, allowing architects and urban planners to predict the evolution of extremely complex systems by the use of artificial intelligence. Further research in this area is necessary, as it is still an emerging area of study. Future developments for this type of methodology include the use of genetic algorithms to simulate the evolution of other urban variables to produce a broader understanding of how a complex self-organizing system, such as a city, will change over time.

In future case studies using this methodology, it should be proposed that a larger sample area be tested since this current study was partially limited by the small size of the population sample (6 buildings). Nonetheless, it can be concluded that the use of evolutionary computation yielded acceptable results when used to predict future urban vertical growth.

10 BIBLIOGRAPHY

Aguirre, H.E., Tanaka, K. and Sugimura, T. (1999). 'Cooperative crossover and mutation operators in genetic algorithms.' Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-99, Morgan Kaufmann, pág. 772, Orlando, Florida, USA.

Akenine, T., Haines, E. and Hoffman N. (2008). 'Real-Time Rendering.' AK Peters / CRC Press (Taylor & Francis).

Alberti, L. B. (1435). 'Della Pittura.' (First appeared 1435-36). Translated with Introduction and Notes by John R. Spencer. New Haven: Yale University Press. 1970 [First printed 1956].

Alberti, L.. B. (1441) 'De Pictura'. Latin version of 'Della pittura.' (First appeared 1439-41). Translated with Introduction and Notes by John R. Spencer. New Haven: Yale University Press. 1970 [First printed 1956].

Almazán, J. and Tsukamoto, Y. (2009). 'Tokyo Public Space Networks at the Intersection of the Commercial and the Domestic Realms (Part III) Study on Transit Urban Centers.' Journal of Asian Architecture and Building Engineering, 8 (2), pp.461-468.

Al-Sayed, K., Turner, A. and Hanna, S. (2012). 'Emergence and self-organization in urban structures.' Proceedings of the AGILE'2012 International Conference on Geographic Information Science, Avignon, April, 24-27, 2012

Andersen, P. and Salomon, D. (2012). 'The Architecture of Patterns.' W.W. Norton and Company, London and New York, 2010.

Anglin, W.S. (1994). 'Mathematics: A Concise History and Philosophy.' Springer, CT, USA.

Angeline, P.J. (1996). 'Two self-adaptive Crossover Operators for Genetic Programming.' Advances in Genetic Programming 2, MIT Press, pág. 89-110. Cambridge, MA, USA.

ASPR (2015). American Society for Photogrammetry and remote Sensing. May 20, 2015, at the Wayback Machine.

Baker, J.E. (1987). 'Reducing Bias and Inefficiency in the Selection Algorithm. Genetic Algorithms and their Applications.' Proceedings of the Second International Conference on Genetic Algorithms, Lawrence Erlbaum Associates, pág. 14-22, Hillsdale, New Jersey, USA.

Baumgart, B. (1975). 'Winged-Edge Polyhedron Representation for Computer Vision.' National Computer Conference, Anaheim, CA, USA.

Bellini, F. (2004). 'Le cupole di Borromini: la "scientia" costruttiva in età barocca.' Electa. Milan, Italy.

Bentley, P. J. (1996). 'Generic evolutionary design of solid objects using a genetic algorithm.' Doctoral dissertation, The University of Huddersfield.

Bernard, F. (2003). 'A history of CATIA.' Dassault Systèmes. France.

Blinn, J. F. (1977). 'Models of light reflection for computer synthesized pictures.' Paper presented at the ACM SIGGRAPH '77 Proceedings of the 4th annual conference on Computer graphics and interactive techniques. July 1977, San Jose, California, USA.

Bishop, G. and Weimer, D. M. (1986). 'Fast Phong shading.' In ACM SIGGRAPH Computer Graphics (Vol. 20, No. 4, pp. 103-106). August 1986, ACM New York.

Booker, L.. B. (1982). 'Intelligent Behavior as an Adaptation to the Task of Environment.' Thesis, University of Michigan, Michigan, USA.

Botsch, M. and Kobbelt, L. (2003). 'High-quality point-based rendering in modern gpus.' Pacific Graphics, pages 335–343, 2003.

Botsch, M., Spernat, M. and Kobbelt, L. (2004). 'Phong splatting.' In Proceedings of the First Eurographics conference on Point-Based Graphics (pp. 25-32). Eurographics Association.

Brindle, A. (1981). 'Genetic Algorithms for Function Optimization.' Doctoral Thesis. Alberta University, Canada.

Bostrom, N. (2014). 'Superintelligence: Paths, Dangers, Strategies.' Oxford University Press, UK.

Burks, A. W. (1960). 'Computation, behavior and structure in fixed and growing automata.' Yovits, M.C. andCameron, S. Editors, Self-Organizing Systems, Pergamon Press, pag. 282-309, New York, USA.

Burry, M. (1999). 'Paramorph: Anti-accident methodologies.' Architectural design: Hypersurface architecture II (pp. 78–83). Perella, S. (Ed.), Chichester, UK.

Burry, M. (2003). 'Between intuition and process: Parametric Design and Rapid Prototyping.' Kolarevic, B (ed) Architecture in the Digital Age: Design and manufacturing. Spon Press, pp. 147-62

Canciani, M., Falcolini, C,. Saccone, M. and Spadafora, G. (2013). 'International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.' Volume XL-5/W1, 2013 3D-ARCH 2013 - 3D Virtual Reconstruction and Visualization of Complex Architectures, 25 – 26 February 2013, Trento, Italy.

Chaos Gropup (2016). http://www.chaosgroup.com/en/2/index.html

Chellapilla, K. (1997). 'Evolutionary programming with tree mutations: Evolving computer programs without crossover.' Genetic Programming 1997: Proceedings of the Second Annual Conference, Morgan Kaufmann, pág. 431-438, Stanford University, CA, USA.

Clarke, K., and Gaydos, L. (1998). 'Loose-coupling a cellular automaton model and GIS: long-term urban growth prediction for San Francisco and Washington/Baltimore.' *Int. J. Geograph. Inf. Sci.,* 12(7), 699–714.

Clarke, K. C., Hoppen, S., and Gaydos, L. (1997). 'A Self-Modifying Cellular Automaton Model of Historical Urbanization in the San Francisco Bay Area.' Environment and Planning B: Planning and Design, 24, 247–261.

Cramer N.L. (1985). 'A Representation for the Adaptive Generation of Simple Sequencial Programs.' Grefenstette J.J. editor, Proceedings of the First International Conference on Genetic Algorithms and Their Applications, Lawrence Erlbaum Associates, pág. 183-187, Hillsdale, New Jersey, USA.

CTBHA (2009). 'Tall Buildings in numbers: World's Tallest Urban Agglomerations.' Council of Tall Buildings and Urban Habitat Journal 2009-Issue 1. Page 1. Chicago. USA.

Darwin, C. (1859) 'On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life.' First published in 1859. John Murray, London, UK.

Davis, D. (2013). 'Modelled on Software Engineering: Flexible Parametric Models in the Practice of Architecture.' Doctoral thesis. RMIT University. Melbourne, Australia.

Dawkins, R. (1987). 'The Blind Watchmaker: Why The Evidence Of Evolution Reveals A Universe Without Design.' W.W. Norton & Company, Inc., New York, USA.

Day, M. (2003). 'Gehry, Dassault and IBM Too.' AEC Magazine.

Delanda, M. (2005). 'Deleuze and the use of genetic algorithm architecture.' Architectural Design V72. John Wiley & Sons, New York, USA.

Delavar, M. R., Naghibi, F., Reza, and Pijanowski, B. (2016). 'Urban Growth Modeling Using Cellular Automata with Multi-Temporal Remote Sensing Images Calibrated by the Artificial Bee Colony Optimization Algorithm.' Sensors (Basel). Vol 16 (12): 2122.

De Jong, K.A. (1975). 'An analysis of the behavior of a class of genetic adaptive systems.' Ph.D. Thesis, Ann Arbor, MI, USA, (1975).

Domingos, P. (2015). 'The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World.' Basic Books, New York, USA.

Dunn, N. (2012). 'Digital Fabrication in Architecture.' Laurence King Publishing. London

Durand, J.N.L. (1805). 'Précis des leçons d'architecture données à l'École polytechnique.' First published in 1805, Paris, France.

Dürer, A. (1525). 'The Painter's Manual: A Manual of Measurement of Lines, Areas, and Solids by Means of Compass and Ruler.' trans. Walter L. Strauss. New York, Abaris Books, 1977).

Eckert, P. (1946). 'The Theory and techniques for Design of Digital Computers.' The Moore School Lectures. University of Pennsylvania. Photorealistic Visualization. Focal Press.

Emporis Building Directory (2017). 'Emporis Building Map: Tokyo.' <https://www.emporis.com/buildings/map> (Accessed June 10th 2017).

Engelen, G., White, R., and Uljee, I. (1997). 'Integrating constrained cellular automata models, GIS and decision support tools for urban planning and policy-making.' *Decision support systems in urban planning*, H. Timmermans, ed., Chap. 8, E&FN Spon, London.

Fogel, L.J. (1964). 'On the organization of Intellect.' Doctoral Thesis, University of California Los Angeles. California, USA.

Friedberg, R.M., Dunham, B. and North J.H. (1959). 'A Learning Machine: Part II.', IBM Journal of Research and Development, vol. 3, pág. 282-287, 1959.

Fuchs, M. (1998). 'Crossover Versus Mutation: An Empirical and Theoretical Case Study.' 3rd Annual Conference on Genetic Programming, Morgan-Kauffman, USA.

Fujiki, C. (1986). 'An Evaluation of Holland's Genetic Operators Applied to a Program Generator.' Doctoral Thesis. Idaho University, Moscow, Idaho, USA.

Gao, Z., Nocera, L., Wang, M. and Neumann, U. (2014). 'Visualizing Aerial LiDAR Cities with Hierarchical Hybrid Point-Polygon Structures.'

Gero, J. S. (1999). 'Evolutionary Design by Computers.' Edited by Bentley, P. Morgan Kaufmann Publishers Inc., San Francisco, USA.

The Global Tall Building Database of the Council of Tall Buildings and Urban Habitat (CTBUH). (2017). 'The Skyscraper Center. Tokyo, Japan.' <http://www.skyscrapercenter.com/city/tokyo> (Accessed June 10th 2017).

Golberg, D. E. (1989). 'Genetic algorithms in search, optimization, and machine learning.' Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

Google Earth (2015). 'Google Maps: Tokyo, Japan.' <https://www.google.co.jp/maps/@35.6471092,139.7591565,433a,35y,318.1h,72.57t/data=!3m1!1e3 > (Data downloaded August 08, 2015).

Goswami, P., Erol, F., Mukhi, R., Pajarola, R. and Gobbetti, E. (2012). 'An efficient multi-resolution framework for high quality interactive rendering of massive point clouds using multi-way kd-trees.' The Visual Computer, Springer.

Guilford, J. P. (1967). 'The nature of human intelligence.' McGraw-Hill, New York, USA.

Hicklin, J.F. (1986). 'Application of the Genetic Algorithm to Automatic Program Generation.' doctoral Thesis. Idaho University, Moscow, Idaho, USA.

Hill, M. (2013). 'Practical and Symbolic Geometry in Borromini's San Carlo alle Quattro Fontane.' Journal of the Society of Architectural Historians 72, no. 4 (December 2013).

Holland, J. H. (1975). 'Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.' U Michigan Press. USA.

Holland, J. H. (1992). 'Genetic Algoritms.' Revista Investigación y Ciencia, pag. 38-45. 1992.

Holland, J. H. (1995). 'Redes de Neuronas Artificiales y Algoritmos Genéticos.' Course at the Computer Science School, Universidade da Coruña, 25th April-11th Mayo 1995. Spain

Holland, J. H. (1998). 'Emergence: From Chaos to Order.' Helix. Massachusetts, USA.

Hurvich, C. M., and Tsai, C. (1990). 'The Impact of Model Selection on Inference in Linear Regression.' The American Statistician, 44(3).

Iwamoto, L. (2009). 'Digital Fabrications: Architectural and Material Techniques.' Princeton architectural Press, p. 5. New York, USA.

Jacobs, J. (1961). 'The Death and Live of the Great American Cities.' Vintage. New York, USA.

Johnson, S. (2001). 'Emergence: The Connected Lives of Ants, Brains, Cities, and Software.' Touch Stone 2001, New York, USA.

Kittler, F. (2001). 'Perspective and the book.' Grey Room 05, Fall 2001, pp 38-58. Grey Room Inc and Massachusetts Institute of technology. USA

Kolarevic, B. (2003). 'Architecture in the Digital Age: Design and manufacturing.' Spon Press, p. 13. New York, USA.

Koza, J. R. (1989). 'Hierarchical Genetic Algorithms operating on Populations of computer programs.' Proceedings of the 11th International Joint Conference on Artificial Intelligence, páginas 768-774. Morgan Kaufmann, San Mateo, California, USA.

Koza, J. R. (1992). 'Genetic programming: on the programming of computers by means of natural selection (Vol. 1).' Massachusetts Institute of Technology. MIT press. Cambridge, Massachusetts, USA.

Koza, J. R. (1996). 'Genetic Programming.' proceedings of the first annual conference, 28-31 July, 1996. Stanford University, California, USA.

Knuth, D. (1998). 'The Art of Computer Programming: Volume 3, Sorting and Searching.' Addison Wesley Longman Publishing Co., Inc. Redwood City, CA, USA.

Krugman, P. (1996). 'The Self-Organized economy.' Blackewell Publishers. Oxford, UK and Malden , Massachusetts.

Krugman, P. (2009). 'The return of Depression on Economics and the crisis of 2008.' Norton & Company Inc. New York. W.W.

Kuder, M., Sterk, M., and Zalik, B. (2013). 'Point-based rendering optimization with textured meshes for fast LiDAR visualization. Computers & Geosciences.' 59(0):181 – 190.

Kudo, M., and Sklansky, J. (1998). 'A comparative evaluation of medium- and large-scale feature selectors for pattern classifiers.' Kybernetika 34(4):429–434.

Kuhlo, M. and Eggert, E. (2010). 'Architectural Rendering with 3ds Max and V-Ray.' Focal Press. Elsevier, Massachusetts, USA.

Leach, N. (2009). 'Digital Morphogenesis.' Architectural Design, Volume 79, Issue 4. John Wiley & Sons, New York, USA.

Levenberg, K. (1944). 'A Method for the Solution of Certain Non-linear Problems in Least Squares.' Quarterly of Applied Mathematics, 2(2):164–168, Jul. 1944.

Levoy, M. and Whitted, T. (1985). 'The use of points as a display primitive.' Technical report 85-022, Computer Science Department, University of North Carolina, at chapel Hill, NC. USA.

Lincoln Laboratory (1964). '*Computer Sketchpad*. Digitized copy of original.' Youtube video, posted 17 November 2007, (https://www.youtube.com/watch?v=USyoT_Ha_bA.).

Leao, S., Bishop, I., and Evans, D. (2004). 'Simulating Urban Growth in a Developing Nation's Region Using a Cellular Automata-Based Model.' Journal of Urban Planning and Development. ASC. Sept 2005.P 145-158.

Li, X., Lin, J., Chen, Y., Liu, X., and Ai, B. (2013). 'Calibrating cellular automata based on landscape metrics by using genetic algorithms.' Int. J. Geogr. Inf. Sci. 27, 594–613

Li, X., Yang, Q., and Liu, X. (2007). 'Genetic algorithms for determining the parameters of cellular automata in urban simulation.' Sci. China Ser. D: Earth Sci. 50, 1857–1866.

Li, X., and Yeh, A. G. O. (2000). 'Modelling sustainable urban development by integration of constrained cellular automata and GIS.' *International Journal. Geographical Information Science,* 14(2), 131–152.

Li, X., and Yeh, A. G. O. (2002). 'Neural-network-based cellular automata for simulating multiple land use changes using GIS.' *International Journal. Geographical Information Science,* 16(4), 323–343.

Liu, H., and Motoda, H. (1998). 'Feature transformation and subset selection.' IEEE Intell Syst Their Appl, 13(2), 26-28.

Lourakis, M.I.A. (2005). 'A brief description of the Levenberg-Marquardt algorithm implemented by levmar.' Technical Report, Institute of Computer Science, Foundation for Research and Technology, Greece.

Luke, S. and Spector, L. (1998). 'A Revised Comparison of Crossover and Mutation in Genetic Programming.' 3rd Annual Conference on Genetic Programming. Morgan-Kauffman.

Machado, P. and Cardoso, A. (2002). 'All the truth about NEvAr.' Applied Intelligence, 16(2), 101-118.

Machado, P. and Cardoso, A. (2000, April). 'NEvAr–the assessment of an evolutionary art tool.' In Proceedings of the AISB00 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science, Birmingham, UK (Vol. 456).

Marcos, C.L. (2010). 'Complexity, Digital Consciousness and Open Form: A New Design Paradigm. Presented on the 30th annual conference of the Association for Computer Aided design in Architecture.' ACADIA 2010, New York.

Marquardt, D.W. (1963). 'An Algorithm for the Least-Squares Estimation of Nonlinear Parameters.' SIAM Journal of Applied Mathematics, 11(2):431–441, Jun. 1963. PA, USA.

Mathias, K., Whitley, L., Stock, C., and Kusuma, T. (1994). 'Staged hybrid genetic search for seismic data imaging.' International Conference on Evolutionary Computation. Orlando, USA, pp. 356-361.

Mauchly, J. (1946). 'The Theory and techniques for Design of Digital Computers.' The Moore School Lectures. University of Pennsylvania. USA.

Meyer, Ulf. (2011). 'Tokyo Architectural Guide.' Dom Publishers. Berlin, Germany.

Miralles, E. and Prats, E. (1991). 'How to layout a croissant.' Enric Miralles y Carme Pinós. En Construcción 1988-1991. El Croquis, 49/50, 252 p. Madrid, Spain.

Monge, G. (1799). 'Géométrie descriptive. Leçons données aux Écoles Normales.' First published in 1799, Paris, France.

Montana, D.J. (1995). 'Strongly Typed Genetic Programming.' Evolutionary Computation, The MIT Press, pag. 199-200, Cambridge, MA, USA.

Nilsson, N. J. (2009). 'The Quest for Artificial Intelligence: A History of Ideas and Achievements.' New York. Cambridge University Press.

Oxman, R. E. (2010). 'Morphogenesis in the theory and methodology of digital tectonics.' Journal of the International Association for Shell and Spatial Structures. 2010, vol. 51, nº 3 (77 p.) pp. 195-205.

Oxman, R. E. (2010). 'Sharing media and Knowledge in Design Pedagogy.' Journal of Information Technology in Construction. ITcon Vol. 15 (2010), Oxman, pg. 291-305.

Pacey, A. (2007). 'Medieval Architectural Drawing.' Stroud: Tempus Publishing. pp. 225–227.

Pazos, R.I. (2014). 'The Tokyo Skyline: Timeline and Morphology.' Journal of Asian Architecture and building engineering. *13*(3), 609-615. Tokyo, Japan.

Pereira, F.B., Machado, P., Costa, E. and Cardoso, A. (1999). 'Graph based crossover-A case study with the busy beaver problem.' Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-99, Morgan Kaufmann, pág. 1149-1155, Orlando, Florida, USA.

Pernice, K. (2004). 'Metabolism Reconsidered: Its Role in the Architectural Context of the World.' Journal of Asian Architecture and Building Engineering, , Vol 3 (2), pp.357-367. Tokyo. Japan.

Piegl, L. and Tiller, W. (1997). 'The NURBS Book.' Monographs in Visual Communication. Springer. Berlin, Germany.

Piegl, L. (1991). 'On NURBS: A Survey.' IEEE Computer Graphics and Applications, Jan 01, 1991, Vol. 11, No. 1, pp. 55 – 71.

Pintus, R., Gobetti, E. and Callieri, M. (2011). 'Fast low-memory seamless photo blending on massive point clouds using a streaming framework.' Journal: Computing and Cultural Heritage (JOCCH) 4(2):6.

Poli, R., and Langdon, W.B. (1998). 'On the ability to search the space of programs of standard, one-point and uniform crossover in genetic programming.' Technical Report CSRP-98-7, Birmingham University, UK.

Quagliarella, D. (1998). 'Genetic algorithms and evolution strategy in engineering and computer science.' Ed. Chichester John Wiley and Sons.

Randall, M.C., Thorne, C.E. and Wild, C. (1994). 'A Standard Comparison of Adaptive Controllers to Solve the Cart Pole Problem.' Proceedings of the Second IEEE Australian and New Zealand Conference on Intelligent Information Systems, pág. 61-65, 1994.

Rienow, A., Stenger, D., and Menz, G. (2014). 'Sprawling cities and shrinking regions – forecasting urban growth in the Ruhr for 2025 by coupling cells and agents.' Erdkunde. Vol. 68 · No. 2 · 85–107

Rocker, I. (2010). 'Interface: Between Analog and Digital Systems.' Presented on the 30th annual conference of the Association for Computer Aided design in Architecture. ACADIA 2010, New York.

Rogers, D. F. (1991). 'State of the Art in Computer Graphics - Visualization and Modeling.' Rae A. Earnshaw (editors), Springer-Verlag, pp. 225 – 269. New York, USA.

Rovira, J., Wonka, P., Sbert, M. and Castro, F. (2005). 'Point sampling with uniformly distributed lines.' Point-Based Graphics 2005, 00, pp. 109–118, (2005).

Rusinkiewicz, S. and Levoy, M. (2000). 'QSplat: A multiresolution point rendering system for large meshes.' In Proceedings of the 27th annual conference on Computer graphics and interactive techniques (pp. 343-352). ACM Press/Addison-Wesley Publishing Co.

Saltmarsh, J. (2015). 'King's College Chapel: A History and Commentary by John Saltmarsh.' Published by Jarrold Publishing.

Schelling, T. (1978). 'Micromotives and Macrobehaviour.' New York and London. WW Norton.

Seber, G. A., & Lee, A. J. (2012). 'Linear regression analysis.' (Vol. 936). John Wiley & Sons.

Seidensticker, E. (2010) 'Tokyo from Edo to Showa 1867-1989: The Emergence of the World's Greatest City.' Tuttle Publishing, p.518. Hong Kong.

Selfridge, O.G. (1958). 'Pandemonium: A Paradigm for learning.' Proceedings of the Symposium on Mechanization of Thought Processes, pág. 511-529, Teddington, UK.

Sharpe, Jason; Lumsden, Charles J; Woolridge, Nicholas (2008). 'In silico: 3D animation and simulation of cell biology with Maya and MEL.' Morgan Kaufmann Martin, p. 263,

Sims, K. (1991). 'Artificial Evolution for Computer Graphics.' ACM Computer Graphics, 25, 319-328, Addison-Wesley: Boston, MA, USA.

Smith, C. (2006). 'On Vertex-Vertex Systems and Their Use in Geometric and Biological Modelling.' Thesis dissertation, University of Calgary, Alberta, Canada.

Soule, T. and Foster, J. A. (1997). 'Code Size and Depth Flows in Genetic Programming.' Genetic Programming 1997: Proceedings of the Second Annual Conference. Morgan Kauffmann. San Francisco, CA. pp 313-320. USA.

Soule, T. (1998). 'Code Growth in Genetic Programming.' Doctoral Thesis, Idaho University, USA.

Stake, R. (2005). 'Qualitative Case Studies.' *The SAGE Handbook of Qualitative Research*, edited Norman Denzin and Yvonnas Lincoln, 443-466. Third edition. Thousand Oaks: Sage. USA.

Strauss, W. (1977). 'The literary remains of Albrecht Dürer.' Translation of and comments to The Painter's Manual by Dürer. Abaris Books, New York, USA.

Sundberg, D. (2011). '8 Spruce Street'. Metals in Construction, Spring 2011.' The Steel Institute of New York. New York, USA.

Sutherland, I. E. (1963). 'Sketchpad: A Man-machine Graphical Communication System.' Doctoral Thesis. Massachusetts Institute of Technology, Lincoln Laboratory. Massachussets, USA.

Sutherland, I. E. (1965). 'The ultimate display.' Proceedings of IFIPS Congress 1965, May 1965, Vol 2, pp. 506-508. New York,USA.

Swetz, F. J. and Katz, V. J. (2011). 'Mathematical Treasures – Gaspard Monge's Descriptive Geometry.' Mathematical Association of America. Convergence

Terzidis, K. (2006). 'Algorithm Architecture.' Oxford: Architectural Press. Elsevier.

Todd, S. and Latham, W. (1992). 'Evolutionary Art and Computers.' Academic Press.

Turing, A. M. (1952). 'The Chemical Basis of Morphogenesis.' Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences, Vol.237, No. 641. (Aug. 14, 1952), pp. 37-72.

Turing, A. M. (1950). 'Computing Machinery and Intelligence.' Mind 49: 433-460

Turing, A. M. (1937). 'On computable numbers, with an application to the Entscheidungsproblem.' The Journal of Symbolic Logic.

Tadao, U. (1995). 'Japanese Civilization in the Modern world: Tourism.' National Museum of Ethnology. Tokyo, Japan.

Utzon, J. (1962). 'Utzon Archives: The Yellow Book.' The New South Wales Archives (NSW). Sydney, Australia.

Vasari, G. (1570). 'Lives of the Most Eminent Painters, Sculptors and Architects.' vol. 1, trans. Gaston Du C. de Vere (London: Medici Society/ Philip Lee Warner, 1912-1914), 494. London. UK.

Weaver, V. (1958). 'A Quarter-Century in the Natural Sciences.' Rockefeller Foundation, New York. USA.

Weinstock, M., Hensel, M. and Menges, A. (2004). 'Emergence: Morphogenic Design.' Strategies, AD. London, UK.

Weinstock, M. and Stathopoulos, N. (2006). 'Advanced Simulation in Design.' Architectural Design, John Wiley & Sons.

Weisberg, D. (2008). 'The Engineering Design Revolution: The People, Companies and Computer Systems that Changed Forever the Practice of Engineering.' (http://www.cadhistory.net/). USA.

Wetzel, A. (1983). 'Evaluation of the Effectiveness of Genetic Algorithms in Combinational Optimization.' Doctoral Thesis. Pittsburgh University. USA.

White, R., and Engelen, G. (1997). 'Cellular automata as the basis of integrated dynamic regional modeling.' *Environ. Plan. B: Plan. Des.,* 24, 235–246.

Wimmer, M. and Scheiblauer, C. (2006). 'Instant points: Fast rendering of unprocessed point clouds.' In PBG (2006). 2, 6.

Woodbury, R. (2010). 'Elements of Parametric Design.' Routledge.

World Bank (2016). 'The World Bank Data: Japan.' <http://data.worldbank.org/country/japan> (Accessed August 08, 2016)

Wu, F. (1998). 'Sim Land: A prototype to simulate land conversion through the integrated GIS and CA with AHP-derived transition rules.' *Int. J. Geograph. Inf. Sci.,* 12(1), 63–82.

Yessios, C. (2003). 'Is There More to Come?.' Architecture in the Digital Age: Design and Manufacturing, edited by Branko Kolarevic, 259–68. Spon Press. New York, USA.

Zwicker, M., Räsänen, J., Botsch, M., Dachsbacher, C. and Pauly, M. (2004). 'Perspective accurate splatting.' Proceedings of Graphics Interface, pages 247–254.

11 ANEX

11.1 Resumen en Castellano.

11.1.1 Introducción y Contextualización.

Los avances tecnológicos en informática y en particular en el campo de representación tridimensional (3D) han conducido recientemente a una revisión metodológica de la representación arquitectónica. Recientes técnicas de 3D permiten una representación exacta de entornos reales y la generación de espacios virtuales, difuminando los así límites entre espacios obtenidos directamente del ‘mundo real’ y espacios ‘no reales generados artificialmente’. Algunos de los últimos desarrollos incluyen tecnologías láser para detectar imágenes y medidas (LIDAR) que permiten el escaneado en 3D de entornos reales para ser posteriormente procesados en un ordenador. Las tecnologías de escáner LIDAR utilizan un rayo láser de alta precisión (dentro de la gama de precisión de 1 mm), capaces de analizar un objeto real o entorno tomando datos de su geometría, color y textura. El formato utilizado por los escáneres suele denominarse nube de puntos, que consiste en una base de datos de puntos referida a un sistema de coordenadas. Cada punto o vértice se define por sus coordenadas (X, Y, Z) que representan las superficies externas de los objetos escaneados. Los puntos también contienen información adicional, tal como el color, el índice de reflejo y los niveles de iluminación. Esta tecnología permite una amplia gama de aplicaciones en arquitectura.

Las ventajas del escaneo 3D LIDAR por otro lado están limitadas por el gran tamaño que suelen tener los escaneos, que hace que sean difíciles o incluso imposible de procesar por los actuales ordenadores personales, limitando su uso. Actualmente se están desarrollando nuevas técnicas de cálculo basadas en puntos, que ya permiten la visualización de millones de puntos capturados por un escaneo LIDAR.

Los sistemas de puntos son también un formato ideal para la aplicación de inteligencia artificial (IA) basada en algoritmos evolutivos, morfogénesis y procesos de auto-organización. El uso de la geometría de puntos en combinación con IA abre la posibilidad de nuevas metodologías de representación y generación arquitectónica, que podrían conducir a cambios importantes en los postulados metodológicos computacionales del diseño arquitectónico y la representación, difuminando así aún más las fronteras entre el 'mundo real' y modelos 3D 'generados artificialmente'.

El uso de computación evolutiva permite la generación de sistemas adaptativos auto-organizados y procesos de morfogénesis geométrica. Turing (1952), en su artículo titulado 'Las bases químicas de la morfogénesis', menciona como existen pautas recurrentes en el crecimiento de las flores, y cómo mediante mecanismos matemáticos los organismos complejos se desarrollan sin directrices centrales. Es importante notar que cuando Turing desarrolló sus fórmulas matemáticas para explicar las pautas recurrentes de morfogénesis en las plantas, idealizó las células como puntos. Las geometrías topológicas fueron más apropiadas y simples para la formulación matemática del crecimiento celular orgánico. Este tipo de crecimiento conduce a unas estructuras específicas que conducen a geometrías recurrentes, pero siempre diferentes, proceso que es similar a algunos sistemas creados por el hombre, como las ciudades. Al Sayed et Turner (2012) han resaltado como los modelos informáticos que simulan el crecimiento urbano, típicamente se han desarrollado basados en mecanismos más característicos de sistemas biológicos, que de sistemas espaciales (modelos de autómata celular). Holland (1975) exploró como reglas simples pueden conducir a un comportamiento complejo, utilizo la lógica de la evolución de Darwin para crear los algoritmos genéticos. Códigos computacionales binarios de unos y ceros y cadenas entrelazadas de ADN constituyen así el genotipo, o código genético. Los comportamientos complejos adquiridos resultante de esos códigos es lo que se denomina fenotipo, como por ejemplo crecer o multiplicarse. Como ha notado Johnson (2001), los procesos evolutivos consisten en la combinación aleatoria de conjuntos de genes, en este caso los códigos, y dependiendo del éxito de las nuevas soluciones y comportamientos, las variaciones resultantes más exitosas pasarán a la siguiente generación. Las mutaciones al azar también juegan un papel clave en el proceso evolutivo, al proporcionar variantes en el sistema. De esta forma los algoritmos genéticos tienen el potencial de ofrecer nuevas posibilidades para la autogeneración informática de geometrías arquitectónicas y urbanas.

11.1.2 Objetivos.

El objetivo de esta investigación es el evaluar experimentalmente el uso de técnicas de representación informática 3D y algoritmos evolutivos, para tanto la visualización 3D y como para procesos de inteligencia artificial, y más en particular morfogénesis, para así lograr una mayor integración entre modelos 3D procedentes del 'mundo real' y modelos 3D 'generados artificialmente'.

La investigación explora el uso de algoritmos evolutivos, aplicándolos a modelos 3D de un objeto arquitectónico o a la geometría de una ciudad, para generar automáticamente procesos de morfogénesis y crecimiento de sistemas. Los parámetros iniciales se introducirán a partir de datos preexistentes, para generar automáticamente posibles geometrías, basadas en las pautas previamente identificadas.

El objetivo final de la tesis es el estudiar el uso de algoritmos genéticos, tanto para la visualización como para la morfogénesis de geometrías 3D, para conseguir mayor grado de realismo en los procesos 3D, mediante el fusionando de objetos 'reales' y 'artificialmente generados'.

11.1.3 Estructura de la Tesis.

La tesis está organizada en capítulos, comenzando con una breve introducción y descripción de los objetivos. Los siguientes tres capítulos describen los conceptos e ideas fundamentales que se utilizarán en el proceso de investigación, la hipótesis y la metodología. El principal trabajo de la tesis doctoral se desarrollará a través de varios casos experimentales que ponen a prueba cada una de las sub-hipótesis. Finalmente, se extraerán conclusiones y se proponen futuras líneas de investigación.

Capítulo 1. Introduce los conceptos, objetivos, antecedentes históricos y la estructura de la tesis.

Capítulo 2. Explora los fundamentos y conceptos básicos que serán utilizados durante el proceso de investigación y para el desarrollo de la tesis.

Capítulo 3. Explica la hipótesis principal de la tesis y las sub-hipotesis.

Capítulo 4. Explica la metodología de investigación utilizada, así como las estrategias que se seguirán para validar la hipótesis.

Capítulo 5. Explora a través de casos experimentales la combinación de datos procedentes de escaneos LIDAR con modelos 3D generados manualmente.

Capítulo 6. Explora la auto-generación de irregularidades y diversidad en modelos 3D, mediante la propuesta de un algoritmo de morfogénesis evolutiva.

Capítulo 7. Propone un algoritmo evolutivo para simular el crecimiento de sistemas complejos auto-organizados, en particular el crecimiento urbano vertical.

El Capítulo 8. Conclusiones finales de la tesis.

Capítulo 9. Propone líneas adicionales de investigación.

Capítulo 10. Bibliografía utilizada para el desarrollo de esta tesis.

11.1.4 Fundamentos

En el capítulo de fundamentos (2) introduce las nociones básicas y conceptos que serán utilizados posteriormente en la tesis para el desarrollo de la investigación y de los casos experimentales. Se describen en detalle los fundamentos relativos a:

- **Métodos de representación digital.** Se explican los fundamentos de representación digital, incluyendo la generación de gráficos 3D, los programas más comunes utilizados para el modelado e infografías 3D, la obtención de datos reales en 3D (estaciones topográficas, fotogrametría y escaneo LIDAR), infografías basadas en puntos (PBR) y los formatos más comunes de modelado 3D (mallas poligonales, NURBS y nubes de puntos).
- **Generación de geometrías curvas complejas.** Se hace una introducción histórica de del uso de geometrías curvas complejas en la arquitectura, y de su relación con los métodos de representación utilizados en el pasado para la generación de dichas geometrías. También se analiza el impacto de la introducción de técnicas de representación en el proceso generativo formal de la arquitectura, analizando el impacto de la introducción de dibujo asistido por ordenadores para la representación y diseño arquitectónico. Se explican también brevemente nuevas técnicas computacionales, introduciendo los conceptos básicos de diseño paramétrico, algorítmico, morfogénesis, sistemas auto-organizativos y la simulación de crecimiento urbano mediante modelos de 'autómata celular'.
- **Inteligencia artificial: computación evolutiva.** Se introducen los conceptos básicos de inteligencia artificial, para luego explicar los fundamentos de la computación evolutiva. Se desarrollan en detalle los algoritmos genéticos y la programación genética, al ser uno de los principales conceptos que serán utilizados en el desarrollo de la tesis. También se hará una descripción

básica de la computación evolutiva interactiva (IEC), y en particular del ‘arte neuro evolucionario’ (NEvAr), concepto que será utilizado para la morfogénesis en 3D.

11.1.5 Hipótesis.

El objetivo de la tesis es utilizar algunas de las tecnologías informáticas actuales más novedosas para reformular las bases metodológicas de la representación y generación arquitectónica.

La principal hipótesis de la tesis es que, mediante la introducción de tecnologías recientes y técnicas de computación avanzadas con un futuro prometedor, se puede lograr un mayor nivel de integración entre modelos 3D obtenidos directamente del ‘mundo real’ y modelos 3D ‘generados artificialmente’. Los métodos y técnicas propuestos contribuirán así a diluir aún más los límites entre los objetos arquitectónicos ‘reales’ y los ‘generados artificialmente’. Los principales conceptos subyacentes en la hipótesis principal de tesis se muestran en la Fig. 42.

Recientes tecnologías de escaneo LIDAR permiten obtener datos 3D del mundo real, de una manera precisa y realista. Técnicas informáticas recientes permiten generar artificialmente modelos de 3D precisos de forma manual. Mediante el uso de inteligencia artificial, los modelos 3D podrían reflejar con mayor precisión condiciones del mundo real como irregularidades, crecimiento o diversidad, que son difíciles de generar de una forma manual. Al combinar estos procesos de obtención de datos reales y la generación artificial de información, se podrían obtener modelos 3D más realistas, no sólo desde el punto de vista visual, sino también desde un punto de vista generativo y morfológico.

La hipótesis principal de la tesis postula que el proceso de difuminar los límites entre objetos 3D extraídos del mundo real y geometrías 3D generadas artificialmente, conllevaría reformulaciones metodológicas en la representación arquitectónica y urbana. La metodología de investigación se centra en tres sub-hipótesis secundarias, que serán estudiadas, desarrolladas y probadas a través de grupos de casos experimentales.

11.1.6 Metodología.

El proceso de investigación se realizará mediante casos experimentales, de los cuales se obtendrán datos, para la posterior evaluación de los resultados obtenidos. El objetivo final es el validar, o no, la hipótesis principal, a través de varios casos experimentales, proponiendo metodologías y algoritmos específicos para cada uno de los casos.

La metodología de investigación utilizada será 'Divide y Conquista' (D&C), la cual se basa en dividir un problema complejo en sub-problemas o partes, hasta que el problema sea lo suficientemente simple como para ser resuelto directamente. Una vez que se encuentren soluciones individuales, se pueden combinar las conclusiones para resolver el problema inicial, en una sola solución. Según Knuth (1998), la idea de usar una lista elementos de elementos para resolver un problema más complejo se remonta hasta la Babilonia del 200 AC., y se ha utilizado durante siglos. Robert Stake (2005) se ha referido al 'estudio de casos colectivos' como a un proceso de elaboración de múltiples estudios, cuya comprensión conduce a un mejor entendimiento, y tal vez a una mejor teorización, de la totalidad.

La tesis pondrá a prueba la hipótesis principal, dividiéndola en las tres sub-hipótesis, que se centrarán en tres sub-temas específicos. Las sub-hipótesis secundarias explorarán la representación, morfogénesis y crecimiento urbano evolutivo, de modelos 3D 'reales' y 'artificialmente generados' respectivamente, con el objetivo final de evaluar la validez de la hipótesis principal.

11.1.7 Visualización de Objetos 3D 'Reales' y 'Artificiales'.

El primer conjunto de casos experimentales se basa en la combinación de objetos obtenidos por medio de escaneados LIDAR 3D del 'mundo real', y objetos 3D 'generados artificialmente'. Pare ello se propondrá un algoritmo de conversión de NURBS a nube de puntos en combinación con PBR.

Los casos experimentales de este capítulo (5) ponen a prueba el uso de geometrías basadas en puntos, en lugar de los formatos más comúnmente utilizados de malla poligonal y NURBS, para determinar de esta manera, si este es un enfoque más eficiente para visualmente combinar lo 'real' y artificial' de cara a la representación arquitectónica y urbana (Fig. 47). En este caso experimental se propone un algoritmo que traducirá los NURBS a formato nube de puntos.

Para ello se utilizará el programa de visualización de nubes de puntos 'ToView', creado por el Departamento de Tecnologías de la Comunicación, de la Facultad de informática de la Universidad de La Coruña. Este programa visualiza puntos como 'splats' or superficies orientadas (cuadrados, discos o elipses) aplicadas a cada uno de los puntos de la nube (Figs. 50 a 52). Para hacer el proceso eficiente se utiliza diferentes niveles de detalle (LOD), haciendo las nubes de puntos más o menos densas, dependiendo de la distancia de la cámara a cada objeto del modelo 3D, consiguiendo así visualización en 3D continua en tiempo real (RT).

El primer sub-caso experimental se basa en la creación de un modelo 3D simple de una casa en formato NURBS. El algoritmo propuesto exportara la geometría NURBS a una nube equidistante de puntos. La casa ya en formato de nube de puntos se inserta en un modelo 3D, el cual procede de un escaneado LIDAR, de una plaza obtenida del 'mundo real'. De esta manera se generan infografías de alta calidad en tiempo real (Figs. 57 a 60), que combinan objetos 3D generados manualmente con objetos 3D escaneados.

Un segundo sub-caso experimental repite el mismo proceso, pero esta vez con un objeto más pequeño de geometría curva muy compleja. Para ello se utiliza un croissant, siguiendo la investigación iniciada por Miralles y Prats (1991) en el que proponen una metodología para dibujar la geometría compleja de un croissant. Un croissant se medirá manualmente y se modelará en 3D, para ser visualizado en un programa de mallas poligonales. Simultáneamente se escanea otro croissant usando tecnología LIDAR, y se visualiza con el programa 'ToView' de infografías basadas en puntos (PBR). Los resultados obtenidos con ambos métodos se comparan para establecer la eficacia de los sistemas de representación basados en puntos en comparación con las mallas poligonales y NURBS (Figs. 62 a 68).

11.1.8 Morfogénesis Evolutiva de Objetos 3D 'Reales' y 'Artificiales'.

El segundo grupo de casos experimentales explorará la creación de diversidad y nuevos diseños en 3D mediante la utilización de morfogénesis evolutiva. Se propone la introducción de técnicas de inteligencia artificial, en particular algoritmos genéticos para la generación automática de geometrías 3D complejas, con el objetivo de crear objetos 3D más reales (Fig. 48).

El primer sub-caso experimental explora la aplicación de algoritmos genéticos para la morfogénesis de geometrías de puntos. Se busca la obtención de diversidad geométrica y la creación de imperfecciones en modelos 3D generados artificialmente, sin tener que modelar manualmente cada muestra. El objetivo es probar mediante el uso de computación evolutiva nuevas técnicas de autogeneración de diversidad en objetos 3D a partir de una muestra original.

Para ello se retoma el último estudio experimental del capítulo anterior, mediante la aplicación del algoritmo de morfogénesis a un objeto de geometría curva compleja: un croissant. El objetivo es generar irregularidades en la superficie (Fig. 74) y diversidad de muestras 3D (Fig. 75). Para ello un croissant real será modelado en 3D, y se transformará en una nube de puntos mediante el algoritmo de conversión propuesto en el capítulo anterior. Se realizará un proceso de morfogénesis a través de un algoritmo evolutivo interactivo, para lograr más realismo geométrico, mediante la generación automática de imperfecciones en su superficie. El algoritmo creará poblaciones, y el usuario elegirá las muestras más satisfactorias para recombinar genéticamente, y así generar de nuevas muestras. El proceso evolutivo se repetirá hasta que los resultados geométricos sean satisfactorios.

El segundo sub-caso experimental probara el uso de computación evolutiva para generar diseños irregulares, basándose en una muestra original y en una serie de parámetros preestablecidos. Para ello se utiliza una metodología similar a la anterior mediante el uso de algoritmos genéticos para realizar un proceso de morfogénesis evolutiva interactiva. Se utiliza como ejemplo de partida la fachada de un rascacielos de 265m de altura, la Beekman Tower (Fig. 81). El objetivo es conseguir un diseño similar mediante el uso de computación evolutiva interactiva (IEC) basándose en el uso de NEvAr, proceso propuesto por Machado and Cardoso (2000 y 2002). Los resultados obtenidos son satisfactorios, y se consigue realizar un diseño similar a la muestra original a partir de superficies planas, generando geometrías curvas complejas mediante el uso de computación evolutiva interactiva (Fig. 83).

11.1.9 Crecimiento Vertical Urbano a Través de Computación Evolutiva

El último caso experimental continua las investigaciones previas sobre la autogeneración de geometrías complejas, pero esta vez se aplica la computación evolutiva a un modelo 3D de un sistema auto-organizado complejo realizado por el hombre: a una ciudad. La idea es predecir y simular el futuro crecimiento vertical urbano, mediante la generación artificial de grupos de rascacielos ('skylines'), que se sobrepondrán sobre un tejido urbano existente (Fig. 49).

Para ello un área en una ciudad es elegida para el proceso de morfogénesis evolutiva vertical, en este caso el distrito de Minato en el centro de Tokio (Fig. 88). Un modelo 3D obtenido mediante fotogrametría del área de estudio se utilizará como base (Figs. 89 a 91). El caso experimental se centrará en edificios que sobrepasen los 130m de altura. El primer paso será identificar y analizar los patrones de crecimiento vertical recurrentes que han generado el área de estudio, para ser posteriormente utilizados como parámetros base para la generación de la nueva geometría. Para ello se realizarán mapas probabilísticos con gradientes (Figs. 93 a 97), que cuantificarán la probabilidad de que nuevos rascacielos sean construidos en un área determinada, basándose en los siguientes parámetros: propiedad del suelo, regulaciones urbanísticas, accesibilidad y factores económicos (Fig. 99).

Los parámetros recurrentes de crecimiento previamente identificados serán usados como base para generar un algoritmo evolutivo (Fig. 101), el cual simulara automáticamente el crecimiento vertical del área de estudio, determinando el número de edificios que serán construidos por año (Tabla 06), y a su vez generará un mapa probabilístico indicando las zonas con mayor probabilidad de que se construyan nuevos edificios en altura (Fig. 102). Los resultados obtenidos por el algoritmo se compararán con el crecimiento vertical observado que se está desarrollando en el área de estudio en un periodo de cuatro años, con el propósito de evaluar la exactitud de los resultados obtenidos (Tabla 06 y Fig. 102).

Posteriormente se estudiarán los parámetros morfológicos de los edificios en altura del área de estudio, en particular su: morfología en planta (Fig. 104), áreas, alturas y esbeltez (Tablas 7 a 11). Los parámetros morfológicos observados se utilizarán como base para un proceso paramétrico de simulación de crecimiento.

Mediante un proceso aleatorio característico de los sistemas auto-organizados se obtendrán simulaciones del nuevo desarrollo urbano, basándose en los resultados obtenidos del algoritmo y en los parámetros morfológicos. Los modelos 3D resultantes combinarán el modelo 3D 'real' de la ciudad con modelos 3D 'generados artificialmente' mediante el proceso evolutivo (fig. 107 a 109). Los resultados obtenidos serán analizados y evaluados, para extraer conclusiones de la tesis.

11.1.10 Conclusiones

Después de la evaluación individual de los resultados obtenidos en cada caso experimental, se deducen primero conclusiones parciales, y luego generales. La hipótesis principal teorizaba que, mediante la introducción de tecnologías desarrolladas recientemente y técnicas avanzadas de computación, se podría lograr un mayor nivel de integración entre los modelos 3D obtenidos directamente del 'mundo real' y modelos 3D 'generados artificialmente' (no existentes en el mundo real). Para probar la validez de la hipótesis principal se propusieron tres algoritmos para cada una de las tres sub-hipótesis, y sus correspondientes casos experimentales:

- **Conclusiones Sub-Hipótesis 1.** Se propuso un algoritmo para la conversión de NURBS y Mallas poligonales en nubes de puntos, el cual fue aplicado y puesto a prueba en un croissant, una casa y una plaza histórica. El estudio experimental propuso una metodología de infografías basadas en puntos (PBR), en combinación con el algoritmo de conversión. Los resultados produjeron representaciones realistas combinando objetos reales escaneados y objetos generados artificialmente en 3D. La calidad de los resultados obtenidos fue fotorrealista, calidad que también se puede lograr con otros softwares de modelado más comúnmente utilizados. Sin embargo, los resultados obtenidos mediante el uso de los algoritmos propuestos son más precisos, ya que provienen directamente de mediciones exactas del 'mundo real'. Futuros desarrollos en PBR harán más eficientes este tipo de técnicas de visualización, siendo lo más probable que se convierta en una herramienta de representación común en arquitectura y diseño urbano, debido a su gran precisión, realismo y ahorro de tiempo que la metodología propuesta conlleva.

- **Conclusiones Sub-Hipótesis 2.** El capítulo 6 propuso un algoritmo genético para generación de diversidad y deformaciones superficiales en modelos 3D de un croissant y un diseño de fachada de un rascacielos. Se obtuvieron resultados satisfactorios en el modelado de geometrías 3D de curvatura compleja mediante el uso de computación evolutiva interactiva. Los algoritmos genéticos fueron capaces de crear la diversidad, a partir de una muestra original, utilizando un proceso evolutivo de selección natural, dentro de unos parámetros preestablecidos. La metodología propuesta ha demostrado ser exitosa en la generación automática de diseños más realistas, mediante el uso de parámetros de diversidad obtenidos directamente del mundo real, posteriormente introducidos en el sistema.

- **Conclusiones Sub-Hipótesis 3.** El capítulo 7, propone el uso de computación evolutiva aplicada a un modelo 3D de una ciudad real, para generar simulaciones de su futuro crecimiento vertical. Los algoritmos genéticos fueron capaces de aprender a partir del análisis de datos históricos del crecimiento vertical del distrito de Minato en Tokio, y replicarlo, con un grado de convergencia del 98%. Al replicar el crecimiento de este sistema auto-organizado, el algoritmo fue capaz de crear modelos 3D que simularon la futura evolución del sistema, generando, asistido por un proceso paramétrico, modelos 3D que simularon exitosamente el crecimiento urbano vertical real.

- **Conclusiones Finales.** En base a la evaluación de los resultados, se puede concluir que el uso de las técnicas de computación evolutiva propuestas, han logrado de forma eficiente y prometedora una mayor integración visual entre modelos 3D obtenidos directamente del 'mundo real' y modelos 3D 'generados artificialmente'. Los procesos de computación evolutiva interactiva también han demostrado lograr un mayor grado de realismo para la creación de geometrías superficiales artificialmente generadas, creando irregularidades y diversidad a través del uso de computación evolutiva. La propuesta de algoritmos genético también ha logrado simular el crecimiento y evolución vertical de complejos sistemas urbanos auto-organizados. En general se puede concluir que el uso de visualización a base de nubes de puntos (PBR) y técnicas evolutivas de inteligencia artificial ha logrado una mayor integración entre modelos 3D obtenidos directamente del 'mundo real' y modelos 3D 'generados artificialmente', tanto desde un punto de vista visual, geométrico como evolutivo.

11.2 Resumo en Galego.

11.2.1 Introducción e Contextualización.

Os avances nas tecnoloxías informaticas, e en particular no campo da representación tridimensional (3D) levou recentemente a unha revisión metodolóxica da representación arquitectónica. Técnicas de 3D recentes permiten unha representación precisa de espazos reais e a xeracion de espazos virtuais, difuminando así as fronteiras entre espazos e obtidos directamente do 'mundo real' e espazos non reais ‘xerados artificialmente'. Algúns desenrolos recentes inclúen tecnoloxías de láser para detectar imaxes e medicións (LIDAR), permitindo o escaneado e a dixitalización de 3D reais para o seu posterior procesamento nun ordenador. As tecnoloxías de escáner LIDAR usan un feixe de láser de alta precisión (dentro do rango de precisión de 1 mm), capaz de analizar un obxecto real ou os datos do ambiente, obtendo a súa xeometría, cor e textura. O formato empregado por escáneres e comúnmente chamado nube de puntos, e consiste nunha base de datos de puntos referida a un sistema de coordenadas. Cada punto ou vértice é definido polas súas coordenadas (X, Y, Z) que representan as superficies externas dos obxectos analizados. Os puntos tamén contén outra información, tales como a cor, o índice de reflexión e os niveis de luz. Estas novas tecnoloxías permite unha ampla variedade de aplicacións na arquitectura.

As vantaxes do escaneo 3D LIDAR, por outro lado estan limitadas polo gran tamaño que xeralmente teñen os escaneos, o que os fai difícil ou mesmo imposíbel de procesar por os ordenadores persoais actuais, estando así o seu uso limitado. Actualmente desenrólasen novas técnicas computacionais baseadas en puntos, que xa permiten a visualización de millóns de puntos capturados por medio do escaneo LIDAR.

Os sistemas de puntos tamén son ideais para a aplicación da intelixencia artificial baseada nos algoritmos evolutivos, morfoxénese e procesos de auto-organización. Usando xeometrías de puntos en combinación con IA, abrese a posibilidade de novos métodos de representación e xeración de arquitectura, o que podería conlevar a grandes cambios nos principios metodolóxicos da proxectacion e representación arquitectónica, difuminando deste xeito inda máis as fronteiras entre modelos 3D do 'mundo real' e os 'xerados artificialmente '.

O uso da computación evolutiva e sistemas adaptativos permite a xeración xeométrica de sistemas auto-organizados e procesos de morfoxénese. Turing (1952), no seu artigo titulado 'As bases químicas da morfoxénese', mencionaba como existen patróns recorrentes no crecemento das flores, e como mediante mecanismos matemáticos organismos complexos crecen sen ninguna directriz central. É importante ter en conta que cando Turing propuso as súas fórmulas matemáticas para explicar os patróns recurrentes da morfoxénese en plantas, as células foron idealizadas como puntos. O uso de xeometrías topolóxicas eran máis axeitado para una formulación matemática sinxela do crecemento celular orgánico. Este tipo de crecemento xenera estruturas específicas mediante xeometrías recorrentes, pero sempre diferentes, un proceso que é semellante a algúns sistemas feitos polo home, como as cidades. Al Sayed et Turner (2012) destacaron como os modelos informáticos que simulan o crecemento urbano, típicamente foron desenvolvidos en base a mecanismos máis característicos dos sistemas biolóxicos, que dos sistemas espaciáis (modelos de autómata celular). Holland (1975) explorou como regras simples poden producir un comportamento complexo, e usou a lóxica da evolución de Darwin para crear os algoritmos xenéticos. Códigos binarios de ordenador formados con uns e ceros forman cadeas de ADN entrelazadas, que constitúen o xenótipo, ou código xenético. Os comportamentos complexos adquiridos derivados destes códigos é o que se chama o fenotipo, como pode ser crecer o multiplicarse. Como notara Johnson (2001), os procesos evolutivos implican a combinación aleatoria de grupos de xenes, neste caso códigos, e dependendo do éxito das novas solucións e comportamentos, as variacións resultantes de maior éxito pasarán á seguinte xeración. As mutacións aleatorias tamén desempeñan un papel fundamental no proceso evolutivo ao ofrecer variacións no sistema. Deste xeito os algoritmos xenéticos teñen o potencial de ofrecer novas posibilidades de auto-xeración computacional de xeometrías arquitectónicas e urbanas.

11.2.2 Obxectivos.

O obxectivo desta investigación é avaliar experimentalmente o uso de técnicas informáticas de visualización 3D e de algoritmos evolutivos, tanto para a o seu uso na visualización en 3D coma para procesos de intelixencia artificial, e máis particularmente procesos de morfoxénese, a fin de lograr unha maior integración entre os modelos 3D obtidos directamente do 'mundo real' e os 3D 'xerados artificialmente'.

A investigación explora o uso de algoritmos evolutivos aplicados a modelos 3D de obxectos arquitectónicos ou á xeometría dunha cidade, para xerar asi automáticamente procesos de morfoxénese e crecemento de sistemas. Os parámetros iniciáis serán introducidos baseados en datos previamente identificados para xerar automaticamente posibles novas xeometrías.

O obxectivo final da tese é o estudiar o uso de algoritmos xenéticos, tanto para a visualización coma para a morfoxénese de xeometrías 3D, e acadar así un maior realismo no procesos de 3D a traveso da fusión de obxectos 'reais' con obxetos 'xerados artificialmente' .

11.2.3 Estructura da Tese.

A tese está organizada en capítulos, comezando cunha breve introdución e descripción dos obxectivos. Os tres capítulos seguintes describen os conceptos e ideas fundamentais que se usarán no proceso de investigación, a hipótese e a metodoloxía. O principal traballo da tese sera desenvolvido a través de varios casos experimentais que examinan cada unha das sub-hipóteses. Por último, se fan as conclusións e futuras liñas de investigacións son propostas.

Capítulo 1. Introduce os conceptos, obxectivos, contexto histórico e estructura da tese.

Capítulo 2. Explora os fundamentos é conceptos básicos que serán utilizados durante a investigación.

Capítulo 3. Explica a principal hipótese da tese e as sub-hipóteses.

Capítulo 4. Explica a metodoloxía e estratexias de investigación a seguir para validación da hipótese.

Capítulo 5. Explorar través de casos experimentais a combinación de datos de xerados por escaneos LIDAR con modelos 3D xerados manualmente.

Capítulo 6. Explorar a auto-xeración de irregularidades e de diversidade en modelos 3D, propoñendo un algoritmo de morfoxénese evolutiva.

Capítulo 7. Propón un algoritmo evolutivo para simular o crecemento de sistemas complexos auto-organizada, en particular o crecemento urbano vertical.

Capítulo 8. As conclusións finais da tese.

Capítulo 9. Propón liñas adicionais de investigación.

Capítulo 10. Referencias bibliograficas utilizadas na tese.

11.2.4 Fundamentos

No capítulo sobre fundamentos (2) se presentan os conceptos e nocións básicas que serán usados máis tarde na tese para o desenvolvemento das investigacións e dos casos experimentais. Descríbense en detalle os fundamentos relativos a:

- **Métodos de representación dixital.** Explícanse os fundamentos da representación dixital, incluíndo a xeración de gráficos 3D, os programas máis comúns utilizados para modelaxe e gráficos 3D, mesura de datos reais 3D (estacións topográficas, fotogrametría e escaneo dixital), infográfias baseadas en puntos (PBR) e os formatos máis comúns de modelaxe 3D (mallas poligonais, NURBS e nubes de puntos).

- **Xeración de xeometrías curvas complexas.** Faise unha introdución histórica do uso de xeometrías curvas complexas na arquitectura, ea súa relación cos métodos de representación utilizados no pasado para xerar esas xeometrías. También analizase o impacto da introdución de técnicas de representación no proceso xenerativo formal da arquitectura, analizando o impacto da introdución de técnicas de deseño e de representación por ordenador. Novas técnicas computacionais tamén son explicadas, introducindo os conceptos básicos de deseño paramétrico, deseño algorítmico, morfoxénese, sistemas de auto-organización e modelos de 'autómata celular' para a simulación do crecemento urbano.

- **Intelixencia artificial: computación evolutiva.** as nocións básicas de intelixencia artificial son introducidas, a continuación, explicanse en detalle os conceptos básicos de computacion evolutiva, algoritmos xenéticos e programación xenética, o ser un dos principais conceptos que serán utilizados no desenvolvemento da tese. Se fará tamén unha descrición básica da computación evolutiva interactiva (IEC), e en particular a 'arte neuro evolutiva' (NEvAr), concepto que se empregará para a morfoxénese en 3D.

11.2.5 Hipótese.

O obxectivo da tese é usar algunhas das últimas tecnoloxías informáticas para reformular os fundamentos metodolóxicos da xeración e representación arquitectónica.

A principal hipótese da tese é que, a través da introdución de tecnoloxías recentes e técnicas avanzadas de computación cun futuro prometedor, pódese acadar un maior nivel de integración entre os modelos 3D obtidos directamente do 'mundo real' e modelos 3D 'xerados artificialmente'. Os métodos e técnicas propostas, poden así, contribuír a diluír aínda máis as diferencias entre os obxectos arquitectónicos 'reais' e os 'xerados artificialmente'. Os conceptos da principal hipótese da Tese móstranse na Fig. 42.

Recentes tecnoloxías de escaneado LIDAR permiten obter datos do mundo real en 3D de forma realista e con alta precisión. Técnicas informáticas recentes permiten xerar artificialmente modelos 3D precisos. Usando a intelixencia artificial, modelos 3D poderán reflexar con máis precisión as condicións do mundo real, como irregularidades, crecemento e diversidade, que son difíciles de xerar nestes intres dun xeito manual. Ao combinar estes procesos para a obtención de datos reais coa xeración artificial de información, podense obter modelos 3D máis realistas, non só dende o punto de vista visual, pero tamén dende o punto de vista morfolóxico.

A principal hipótese da tese postula que o proceso de diluir os límites entre obxectos 3D procedentes do ‘mundo real’ con xeometrías 3D ‘artificialmente xeradas’, conlevara a reformulacions metodolóxicas na representación arquitectónica e urbana. A metodoloxía de investigación céntrase en tres sub-hipóteses secundarias, que serán desenvolvidas e probadas en grupos de casos experimentais.

11.2.6 Metodoloxía.

O proceso de investigación se realiza mediante procesos experimentais, dos cales serán obtidos datos para a posterior avaliación dos resultados. O obxectivo final é o validar, ou non, a hipótese principal, a través de varios casos experimentais, para elo proponse metodoloxías e algoritmos específicos para cada un dos casos.

A metodoloxía de investigación usada é 'dividir e conquistar' (D& C), que se basea na división dun problema complexo en sub-problemas ou partes ata que o problema é suficientemente sinxelo para ser resolto directamente. Unha vez encontranse as solucións individuais, podense combinar para resolver o problema inicial, nunha única solución. Segundo Knuth (1998), a idea de usar unha lista de elementos individuais para resolver un problema máis complexo remontase á Babilonia do 200 AC., e foi usada dende hai séculos. Robert Stake (2005) referiuse ao 'estudo colectivo de casos' como un proceso de facer varios estudos, cuxo entendemento leva a unha mellor comprensión, e quizais mellor teorización da totalidade.

A tese probara a hipótese principal, dividíndoa en tres sub-casos, e enfocados en tres sub-temas específicos e sub-hipóteses secundarias: a representación dixital, a morfoxénese xéometrica eo crecemento urbano evolutivo, de modelos 3D 'reais' e modelos 'xerados artificialmente', co obxectivo final de avaliar a validez da hipótese principal.

11.2.7 Visualización de Obxectos 3D ‘Reais' e 'Artificiais.

O primeiro conxunto de casos de proba está baseado na combinación de obxectos obtidos por medio de escaneos 3D LIDAR do 'mundo real' e obxetos 3D 'xerados artificialmente'. Para elo proponse un algoritmo de conversión de nube de puntos a NURBS en combinación con PBR.

Os estudos experimentais neste capítulo (5) pon a proba o uso de xeometrías a base en puntos, en vez das xeometrias de malla poligonal e NURBS, que son máis comunes, para así determinar se este é un método máis eficaz para fundir visualmente o 'real' é ‘artificial’, na representación urbana e arquitectónica (Fig. 47). Neste caso experimental un algoritmo que pode traducir NURBS a formato de nube de puntos é proposto.

Para iso, será usado o programa de visualización de nubes de puntos ‘ToView', creado polo Departamento de Tecnoloxías da Comunicación, da Facultade de Informática da Universidade da Coruña. Este programa amosa os puntos como ‘splats’ ou superficies orientadas (cadrados, discos ou elipses) aplicadas a cada un dos puntos da nube (Figs. 50 a 52). Para facer o proceso eficiente diferentes niveis de detalle (LOD) son usados, facendo nubes con máis ou menos puntos, dependendo da distancia a cámara de cada obxecto no modelo 3D, conseguindo así unha visualizacion continua do 3D en tempo real (RT).

O primeiro caso sub-experimental baséase na creación dun modelo 3D dunha casa simple en formato NURBS. O algoritmo proposto exportara a xeometría NURBS coma unha nube de puntos equidistantes. A casa 3D xa en formato nube de puntos será insertada nun modelo 3D, procedente dun escaneo LIDAR, dunha praza do 'mundo real'. Xeneranse asi visualizacions de alta calidade en tempo real (Figs. 57 a 60), que combinan obxectos 3D xerados manualmente con obxetos 3D escaneados.

Un segundo sub-caso experimental repite o mesmo proceso, pero esta vez con un pequeno obxecto de moi complexa xeometría curva. Para iso usase un croissant, seguindo a investigación iniciada por Miralles e Prats (1991), que propón unha metodoloxía para deseñar a xeometría complexa dun croissant. Un croissant mídese a man primeiro, e despois é modelado en 3D no formato malla polígonal. Simultaneamente outro croissant escanease con LIDAR, para ser visualizado co programa 'ToView' baseado en puntos (PBR). Os resultados obtidos cos dous métodos son comparadas para determinar a eficacia dos sistemas representativos de nubes de puntos comparados coas mallas poligonais e NURBS (Figs. 62 a 68).

11.2.8 Morfoxénese Evolutiva de Obxectos 3D 'Reais' e 'Artificiais'.

O segundo grupo de casos experimentais explora a creación de diversidade e de novos deseños 3D usando morfoxénese evolutiva. Proponse a introdución de técnicas de intelixencia artificial, incluíndo algoritmos xenéticos para a xeración automática de xeometrías complexas en 3D, co obxectivo de crear obxectos 3D máis reais (Fig. 48).

O primeiro sub-caso experimental explora a aplicación de algoritmos xenéticos a xeometrías de puntos para producir morfoxénese. Buscase a obtención de diversidade e imperfeccións xeométricas en modelos 3D xerados artificialmente, sen ter que modelar manualmente cada mostra individualmente. O obxectivo é auto-xerar mediante o uso de técnicas de computación evolutiva diversidade nos obxectos 3D a partir dunha mostra orixinal.

Para elo retomase o derradeiro estudo experimental do capítulo anterior, aplicando o algoritmo de morfoxénese unha xeometría curva dun obxecto complexo: un croissant. O obxectivo é xerar diversidade de mostras (Fig. 75) e irregularidades na superficie (Fig. 74). Un croissant real será modelados en 3D, e transformado nunha nube de puntos mediante o algoritmo de conversión proposto no capítulo anterior. Un proceso de morfogese realizarse por medio dun algoritmo evolutivo interactivo, para conseguir máis realismo xeométrico través da xeración automática de imperfeccións na súa superficie. O algoritmo creará poboacións de croissants, o usuario escollerá as mostras máis satisfactorias para recombinarlas xeneticamente e xerar novas mostras. O proceso evolutivo será repetido ata que os resultados xeométricos sexan satisfactorios.

O segundo caso sub-caso experimental probara a computación evolutiva para xerar patróns irregulares, baseandose nunha mostra orixinal e nunha serie de parámetros prestablecidos. Para este fin un método semellante ó anterior é utilizado mediante o uso de algoritmos xenéticos para realizar un proceso de morfogese evolutiva interactiva. Coma exemplo utilizase a fachada dun rañaceos de 265m de altura, a torre de Beekman (Fig. 81). O obxectivo é conseguir un deseño similar, pero utilizando computación evolutiva interactivo (IEC) baseada na utilización de 'NevAr', proceso proposto por Machado e Cardoso (2000 e 2002). Os resultados obtidos son satisfactorios, e consiguese facer un deseño semellante á mostra orixinal, a partir dunha superficies plana, creando xeometrías curvas complexas mediante o uso da computación evolutiva interactiva (Fig. 83).

11.2.9 Crecemento Vertical Urbano a Través da Computación Evolutiva

O último caso de proba continuoa as investigacións anteriores sobre a auto-xeración de xeometrías complexas, pero agora a computación evolutiva será aplicada a un modelo 3D dun complexo sistema auto-organizado feito polo home: unha cidade. A idea é e simular o futuro crecemento vertical urbano a través da xeración artificial de grupos de rañaceos ('skylines'), para ser logo sobreposta ó un texido urbano existente (Fig. 49).

Unha zona dunha cidade é escollida para o proceso de morfoxénese evolutiva vertical, neste caso, o distrito de Minato no centro de Toquio, no Xapón (Fig. 88). Un modelo 3D obtido por fotogrametría será utilizado coma base (Figs. 89-91). O caso de proba centrarase en edificios que superen os 130m de altura. O primeiro paso é identificar e analizar os patróns repetitivos de crecemento verticais que xeraron a área de estudo, para ser posteriormente usados como parámetros base para xerar a nova xeometría. Para elo faranse mapas probabilísticos con gradientes (figuras 93 a 97), os cales cuantifican a probabilidade de que novos rañaceos sexan construídos nunha área determinada, baseandose nos seguintes parámetros: propiedades do solo, regulamentos urbanos, accesibilidade e factores económicos (Fig. 99).

Os parámetros recorrentes de crecemento previamente identificados serán utilizados coma base para xerar un algoritmo evolutivo (Fig. 101) que simulará automáticamente o crecemento vertical da área de estudo, determinando o número de edificios que serán construidos por ano (Táboa 06), e á súa vez, xenerará un mapa probabilístico indicando as zonas máis propensas para que novos edificios altos sexán construídos (Fig. 102). Os resultados obtidos polo algoritmo serán comparados co crecemento vertical observado na área de estudo ao longo dun período de catro anos, a fin de avaliar a precisión dos resultados obtidos (táboa 06 ea Fig. 102).

Posteriormente estudiaranse os parámetros morfolóxicos dos edificios en altura da área do estudo, en particular: a morfoloxía da planta (Fig. 104), as áreas, alturas e esbeltez (Táboas 7 a 11). Os parámetros morfolóxicos observados, utilizaranse como a base para un proceso de simulación paramétrico de crecemento.

Usando un proceso aleatorio característico dos sistemas auto-organizados simularase o novo desenvolvemento urbano, baseandose nos resultados do algoritmo e os parámetros morfolóxicos. Os modelos 3D resultantes combinarán o modelo da cidade 'real' con un modelo 3D 'xerado artificialmente' mediante un proceso evolutivo (Fig. 107-109). Os resultados serán analizados e avaliados, para extraer as conclusións da tese.

11.2.10 Conclusións

Tras a avaliación individual dos resultados experimentais obtidos en cada caso, primeiro dedúcense as conclusións parciais, seguidas das xerais. A principal hipótese da teses é que a través da introducción de novas tecnoloxías e técnicas de computación avanzadas, poderiase acadar un maior nivel de integración entre modelos 3D obtidos directamente do 'mundo real' e modelos 3D 'xerados artificialmente' (existentes no mundo real). Para así probar a validez da hipótese principal. Para elo foron propostos tres algoritmos para cada unha das tres sub-hipóteses e os correspondentes casos experimentais:

- **Conclusións Sub-Hipótese 1.** Foi proposto un algoritmo para converter NURBS e mallas poligonais a nubes de puntos, foi implantado e probado nun croissant, nunha casa e nunha praza histórica. O estudo experimental propuso unha metodoloxía de infografías baseadas en puntos (PBR), en combinación cun algoritmo de conversión. Os resultados produxeron representacións realistas combinando obxectos reais escaneados con obxectos 3D xerados artificialmente. A calidade dos resultados obtidos foi fotorrealística, calidade que tamén pódese conseguir mediante a modelaxe manual con programas máis comúnmente utilizados. Por outra banda, os resultados obtidos mediante os algoritmos propostos son máis precisos porque se trata de medicións precisas do 'mundo real'. Desarrollos futuros nas técnicas PBR farán máis eficiente este tipo de técnicas de visualización, e o máis probable é que co tempo convertanse nunha ferramenta de representación común na arquitectura eo urbanismo, debido á súa alta precisión, realismo e o aforro de tempo que a metodoloxía proposta implica.

- **Conclusións Sub-Hipótese 2.** O capítulo 6 propón un algoritmo xenético para a xeración de diversidade e deformacións superficiais en modelos 3D dun croissant e no deseño dunha fachada dun rañaceos. Obtivéronse resultados satisfactorios no modelado 3D de xeometrías curvas complexas usando computación evolutiva interactiva. Os algoritmos xenéticos foron capaces de crear diversidade a partir dunha mostra orixinal, utilizando un proceso evolutivo de selección natural, dentro duns parámetros predefinidos. A metodoloxía proposta foi exitosa na xeración automática de modelos máis realistas, utilizando parámetros de diversidade obtidos directamente do mundo real.

- **Conclusións Sub-Hipótese 3.** O Capítulo 7, propón a utilización de computación evolutiva aplicada a un modelo 3D dunha cidade real, para xerar deste xeito simulacións de crecemento vertical futuro. Os algoritmos xenéticos foron capaces de aprender a partir da análise do crecemento vertical e en datos históricos do distrito de Minato en Toquio, e replica-lo con un grao de converxencia do 98%. Ao replicar o crecemento deste sistema auto-organizado, o algoritmo foi capaz de crear modelos 3D que simularon a evolución futura do sistema, xerando, asistida por un proceso paramétrico modelos 3D que simularon con éxito o crecemento urbano vertical real.

- **Conclusións Finais.** Baseandose na avaliación dos resultados, pódese concluír que o uso das técnicas propostas de computación evolutiva conseguiron de forma eficaz e prometedora unha maior integración visual entre modelos 3D obtidos directamente do 'mundo real' e modelos 3D ‘xerados artificialmente'. Procesos de computación interactiva evolutiva tamén demostraron poder acadar un maior grao de realismo para crear xeometrías superficiais artificialmente, xenerando irregularidades e diversidade a través da utilización da computación evolutiva. Os algoritmos xenéticos propostos tamén conseguiron simular o crecemento e a evolución vertical de sistemas urbanos auto-organizados complexos. En xeral, pódese concluir que o uso da visualización baseada en puntos (PBR) e técnicas de intelixencia artificial evolutivas foi capaz de acadar unha maior integración entre os modelos 3D obtidos directamente do 'mundo real' e modelos 3D 'xerados artificialmente', tanto dende un punto de vista visual, xeométrico e evolutivo.