ISSN 1828-5961

Modeling singular neoclassical spaces in a procedural way

This research article aims to propose a new method for procedurally modeling unique architectural heritage spaces and explore new approaches to 3D-model parameterizable architecture, specifically focusing on neoclassical architecture.
Procedural modeling, known for its time-saving benefits, has been used in CGI and architectural drawing alongside parametric modeling. The study applies these concepts to the field of architectural heritage digital modeling, which presents challenges due to deformations and imperfections that need to be parametrized.
The research focuses on the side chapels of the Cathedral of Valencia, built in the neoclassical style, and develops an adaptable parametric model to account for their slight dimensional differences.
The methodology involves using interior point cloud data from a 3D scan of the cathedral, creating an ideal model, dividing it into parts, and developing graphic scripts using Grasshopper for Rhinoceros.
The research yields a new pipeline for 3D modeling parameterizable architecture, saving time and resulting in accurate digital models of the side chapels. The study concludes that procedural and parametric modeling can be used not only for projected architecture but also for surveying existing architecture and heritage. The proposed workflow reduces the time required to model similar architectural cases while maintaining accuracy.

Eduardo Baviera Llópez
Architect. He combines his professional work with research in architectural graphic expression. He is currently developing his doctoral thesis on new methodologies for the survey and graphic reconstruction of the neoclassical phase of the Cathedral of Valencia.

Jorge Llopis Verdú
Phd. Architect and Full Professor at the Universitat Politècnica de València in the Architectural Graphic Expression Department. His research lines are focused on the analysis of documentary and cartographic information on architectural heritage and on the study of new architectural graphic strategies after the arrival of digital drawing.

Ignacio Cabodevilla-Artieda
PhD. Architect and Adjunct Professor at the School of Architecture, and a member of the Color Research Group in Architecture (GICA) of the Heritage Restoration Institute at the Universitat Politècnica de València (UPV). His main research lines include the Architectural Heritage of the former Crown of Aragon as well as color in architecture, both historical and modern.

Keywords:
Architectural survey; procedural modelling; parametric modelling, neoclassical architecture; Grasshopper

DOI: https://doi.org/10.20365/disegnarecon.30.2023.16

ISSN 1828-5961

1. INTRODUCTION

Procedural modelling has been considered as a time saving alternative to manual modelling since the dawn of CGI (Ebert et al., 2002). Its application in architectural drawing has come hand in hand with parametric modelling (Woodbury et al., 2011) chasing ultimately the idea of reaching an automated architectural design generator (Rodrigues, 2014).

Parametric modelling is a computer design paradigm that treats the geometric properties of the designed object as variables. The dimensions, angles and, in general, any measurable geometric property (such as curvature) remain modifiable as the definition of the model advances (Schumacher, 2014) by programming geometrical algorithms. Therefore, once the model is finished, there is a general model (genotype) whose parameters can be modified to generate an endless number of different versions (phenotypes).

The piece of work that is being presented aims to apply all these ideas to our field of research, that is, the architectural heritage digital modelling. Architectural heritage may present deformations occurred all over time that can be parametrized, such as flexing or lacking verticality. Some other imperfections are due to the construction process itself, so we may find small metric deviations when comparing two spaces that were supposed to be built following the same -or similar- planimetrics. In this sense, the task of the digital modeller may be quite tedious for sometimes it is important to model these singularities, which prevents the copy-paste method from being used (L. Cipriani et al., 2019).

Our research focuses on the architectural analysis and parametric modeling of the lateral chapels [1] of the Cathedral of Valencia. These chapels, constructed in the neoclassical style during the 18th century, form part of a larger transformation that encompassed the concealment of the cathedral's previous architectural phases, namely the original Gothic structure with some Renaissance interventions (Cortés Meseguer, 2014). Although these chapels share a similar typology, their slight dimensional variations require a unique approach to modeling rather than a simple copy-and-paste solution.

To address this challenge, we propose an adaptive parametric modeling workflow. By leveraging the capabilities of parametric design, we aim to reduce the workload involved in modeling each chapel individually. Instead of manually reproducing the architectural elements for each chapel, our workflow enables the creation of an ideal parametric model that can be adjusted to accommodate the specific dimensions and characteristics of each chapel. This approach allows for efficient iteration and customization, ensuring that the parametric models accurately represent the unique features of each chapel while maintaining a consistent design language.

By implementing this workflow, we aim to streamline the modeling process, saving time and effort for researchers, architects, and designers involved in the study and preservation of these significant architectural structures.

Fig. 1 - View of a lateral chapel of the epistle side

DOI: https://doi.org/10.20365/disegnarecon.30.2023.16

ISSN 1828-5961

2. METHODOLOGY

The starting point of our workflow is, on one hand, the interior point cloud of every chapel coming from a more complete 3D scan work of the whole cathedral (Fig. 3). This scan work was performed with the scanner FARO Focus3D 130 and registered using SCENE sofware.

To achieve a comprehensive interior survey, a total of 100 stations were employed, configured at a high-resolution level, meaning that at a distance of 10 meters from the scanner, we obtained a point matrix with a spacing of 6.3 x 6.3 millimeters. All stations were completed in full rotation, utilizing the scanner's maximum field of view (360º horizontally, 310º vertically). Specifically, for the chapels, one station was positioned at the central point of their floor plan, and another at the center of the threshold beneath the entrance arch. Consequently, due to point cloud overlap, it can be determined that the point density exceeds the aforementioned description.

Fig. 3 - Point cloud model of the Cathedral's inner space

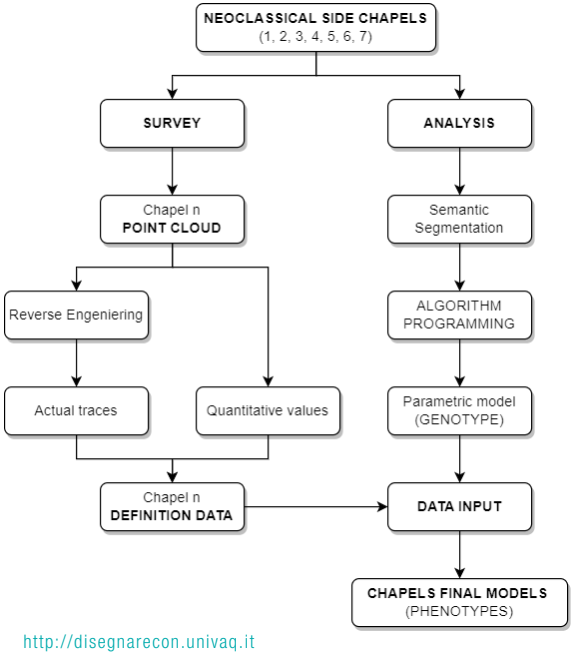

Fig. 2 - General pipeline of our method

The point cloud data for each chapel is exported in E57 format and prepared for importation into reverse engineering software, such as Geomagic Design X.On the other hand, analysing the geometry of the neoclassical chapels (Fig. 1) it is possible to find out some common composition rules and guidelines for all of them as if they were created from an ideal perfect chapel which was metrically adapted according to certain parameters. Our aim at this point is to define those geometrical rules.

DOI: https://doi.org/10.20365/disegnarecon.30.2023.16

ISSN 1828-5961

SURVEY

Every chapel's point cloud is imported into a reverse engeniering software [2], separately. When deemed necessary, the quality of the point cloud can be enhanced by removing noise, which refers to undesirable points resulting from instrumentation errors or unwanted physical objects detected by the scanner, such as pieces of furniture or visitors in the cathedral. For this purpose, we have tools that allow us to eliminate noise automatically and manually. The "filter noise" command enables us to remove points and point clusters that are created due to scanner measurement errors. For example, in cornices, the scanner may place false points on a plane defined by the upper edge and the scanner's center, creating the illusion of an inclined plane. These points are easily identifiable because they belong to less dense clusters. By using the filter noise tool, we can set a threshold for the maximum number of points belonging to a cluster. Clusters of points are determined by their average distance, so any clusters that do not exceed the set threshold will be eliminated. Unfortunately, not all noise can be suppressed using this tool. In such cases, we must resort to manual procedures by utilizing advanced point selection tools to remove them.
Our current objective is to obtain from the point cloud model the traces or guidelines necessary for our geometric algorithm to accurately model the corresponding architectural element. In this regard, the traces we aim to obtain consist essentially of a series of polygonal and curved lines that define the geometry of the architectural elements to be modeled through both horizontal and vertical sections. To achieve this, prior to obtaining each trace, we need to define the plane in which it lies. Subsequently, using the "sketch" tool, we intersect the plane with the point cloud, resulting in a section line with a relatively organic geometry (depending on the precision of the point cloud). We then draw the traces rationally on this section line. Geomagic provides us with a set of two-dimensional design tools similar to those found in any CAD program, including line, arc, circle, ellipse,

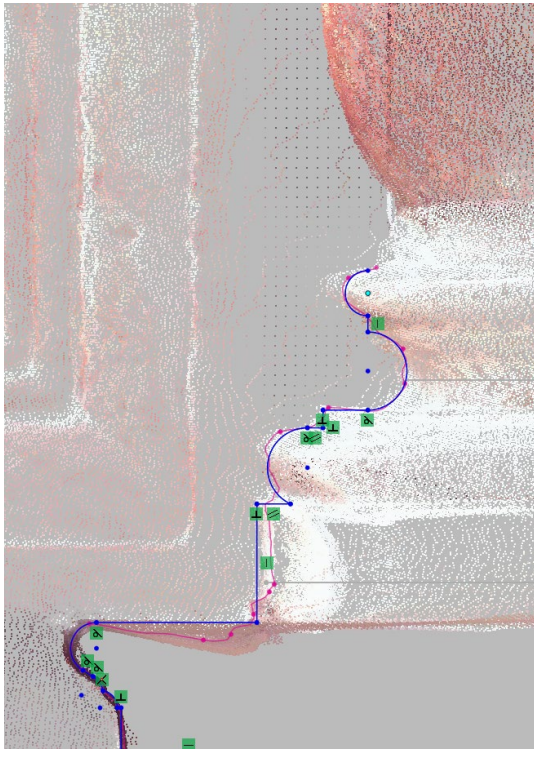

Fig. 4 - Example of a base profile trace

extend, trim, reflect, etc. The uniqueness of this software lies in its ability to allow us to draw pure geometry on the automatic section line using a best-fitting algorithm. For example, a horizontal section line of a wall will never be a perfectly straight line due to execution and material imperfections. However, when drawing rationally, we want to represent it as a perfect straight line. With the line tool activated, we select the section line segment appropriately, and the program adjusts the line to best match the selection. This can be done for straight lines as well as for circles and arcs.
Furthermore, this software allows us to impose constraints on the geometry. These constraints can affect each object individually (horizontal, vertical, inclination angle, fixed length, fixed radius, etc.) or, when combining multiple lines and curves, their relationships with each other (tangent to, perpendicular to, parallel to, etc.). In this way, we can create adjustable parametric profiles by setting some constraints and modifying the remaining parameters (Fig. 4).
We need to obtain two types of traces: general traces that define the dimensions and general shape of the main volumes delimiting the interior of the chapels, and detailed traces that define the precise shape of the moldings of different architectural elements (bases, cornices, surrounds, etc.).

Once we have obtained the necessary traces to define the geometry of the chapel, we export the lines in DXF format and import them into Rhino 3D. It is advisable to redraw the imported profiles, paying special attention to curved parts, as they are imported as polylines with multiple straight segments. After completing the redrawing of the traces, we are ready to begin the actual process of parametric modeling.

ANALYSIS

To approach this process, the first step is to conceptually segment the chapels, following a semantic logic (Benedetti et al., 2010), into the different architectural elements that compose them. This allows us to establish which geometric algorithms are necessary to compose the general procedural model. The established categories are:
1. Pedestals
2. Pilasters
3. Walls
4. Entablature
5. Lunettes
6. Drum
7. Dome
8. Pendentives
9. Pediment (access)
10. Central altar
11. Side altars

DOI: https://doi.org/10.20365/disegnarecon.30.2023.16

ISSN 1828-5961

With the aforementioned categories defined, we can establish an efficient working scheme: first, an independent geometric algorithm is programmed for each segmented architectural element, and second, a general algorithm is programmed to articulate and compose all the partial algorithms to form the overall model of the chapel. The programming of these algorithms is carried out in Grasshopper for Rhinoceros [3]. In Grasshopper, the network of relationships is established and visually displayed, allowing the designer to follow and modify it during the design process.

The definition of design algorithms in Grasshopper is achieved through visually programmed scripts, in contrast to traditional coding through written code [4]. This is done using a system of nodes [5]: functional blocks are added to the Grasshopper "canvas" and connected to each other by "wires" to generate a graphical script called a "definition," which is saved in a GHX file format. Functional blocks or components are characterized by having one or more inputs (on the left side of the block) and outputs (on the right side of the block) where wires connect to receive and send information. Additionally, multiple GHX files can be interconnected to work together within a single model. This allows us to divide the computational load by executing scripts in parts and organize the project structure in a more orderly and intelligible manner, facilitating debugging.

There are two types of blocks: parameters and components. Parameters store data, such as numbers, colors, or geometric objects, among others. Parameters are container objects typically represented as small rectangular boxes with a single input and output. Those that store geometry (points, curves, planes, etc.) can reference it from Rhino or inherit it from other components. "Input parameters," such as a numeric slider or a graph mapper, are dynamic interface objects that enable the designer to interact with the definition. On the other hand, components perform actions based on the input data they receive. There are many types of components for various tasks, including mathematical and vector operators, geometry constructors, and more (Molina-Siles, 2016).
Regarding the data handled by the components, two types can be distinguished: persistent and volatile. Persistent data are specifically set by the designer and do not automatically change with each script update. Volatile data is inherited from one or more sources (components) and, as the name suggests, is not permanent and is destroyed each time the script is restarted. Generally, most data generated in the program flow is considered volatile.

Grasshopper definitions, or graphical scripts, have a Program Flow that represents where the program execution begins, what happens in the middle, and how to determine when the program execution is complete. Visual programs are executed from left to right, allowing the graphical script to be read by following the connections through wires (following the direction of "current") to understand the logic in executing actions. Therefore, all objects and the cables that connect them graphically represent the program logic. This graphical system displays the program flow and the dependency relationships between different parts based on their relative position. Whenever the graph is modified at any point, the connections and objects downstream in the program flow are updated.

At this point, it is worth explaining the approach to the file system for the parametric modeling of the chapels. The general idea is to create a series of visual programs, i.e. Grasshopper definitions, that parameterize the modeling of each architectural element of the chapels, allowing it to adapt to the particular characteristics of each case. These definitions should work coherently to form the complete parametric model of each chapel.
In theory, it would be possible to create a single definition that parametrizes all architectural elements within it. However, Grasshopper's computational capacity and virtual memory management are limited, making it very difficult to execute such a complex program. It should be noted that

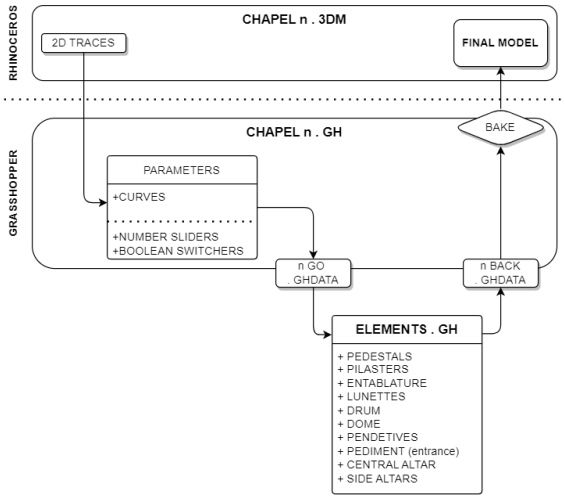

Fig. 5 - General scheme of work

each change made to a parameter or component triggers the script to be executed again, updating the components downstream in the program flow. Furthermore, a definition of such magnitude would be so large that it would hinder the designer's (or developer's) understanding and, consequently, complicate the debugging process of potential errors (also known as bugs) in the script.

Therefore, we create a base file in Rhino for each of the chapels (chapel n.3dm). This document contains the imported two-dimensional traces from Geomagic, which have been redrawn in Rhino. Associated with this file, we create the general definition of the chapel n in Grasshopper (chapel n.gh). This is the graphical script that relates the parametric models of each architectural element and coherently displays them in the associated Rhino base file, forming the complete chapel model.

The general definition file of the chapel contains all the input parameters that can be modified to

DOI: https://doi.org/10.20365/disegnarecon.30.2023.16

ISSN 1828-5961

adjust the model to each case. We use only three types of parameters: curves, numeric sliders, and boolean toggles. Curves allow us to incorporate the traces from the base file into the script. Numeric sliders are used to set a value and define dimensions (height, depth, rotation, etc.). It is possible to establish a range (a maximum and minimum value) as well as a decimal precision to accurately adjust the desired value. Boolean toggles have two positions, true and false. They allow us to activate or deactivate both individual component functions and specific parts of the program. For example, using a boolean toggle, we can inform the program whether the chapel in question has side altars (true) or does not have them (false).

As we can see, in the main definition of the chapel, the entry of the model-defining parameters into the system is managed. Once collected, they are "sent" to the corresponding definitions of each architectural element through data output components. The architectural elements have been classified into the previously mentioned eleven semantic categories (pedestals, pilasters, walls, etc.).

An additional category is added, the floor, which is simply a horizontal plane at elevation 0 with specific dimensions and can be activated or deactivated at will (Fig. 7).

Within each of these categories, there may be one or more definitions depending on the uniqueness of the architectural elements they contain. For example, within the pilasters category, five different definitions have been developed for the five types of pilasters found in the chapels. In addition to these, a sixth definition has been created specifically for placing the capitals on their respective pilaster shafts.

As mentioned, the data output components connect the input parameters with the corresponding definitions, where the data is received through input components. The scripts are executed there, generating models of the architectural elements. These models are then sent back to the main definition following the same process of data output and input.

In the output component, a field is created for each parameter to be transmitted, along with an input to connect the information. The dataset is exported and stored in GHDATA file format. Subsequently, in the corresponding definition, this information is inserted using an input component that references the aforementioned file and adopts the same number of fields as its output counterpart. For each field, a cable output is generated to complete the transmission of the parameter set in the general definition. In order to maintain order in the methodology, the suffix "_go" is added to the GHDATA files that emit information (the input parameters) from the general definition, while the suffix "_back" is added to those files that import the results back from the definitions of the different elements. For example, the parameter emission file of the dome definition is "Dome_go.ghdata," and the result import file is "Dome_back.ghdata."

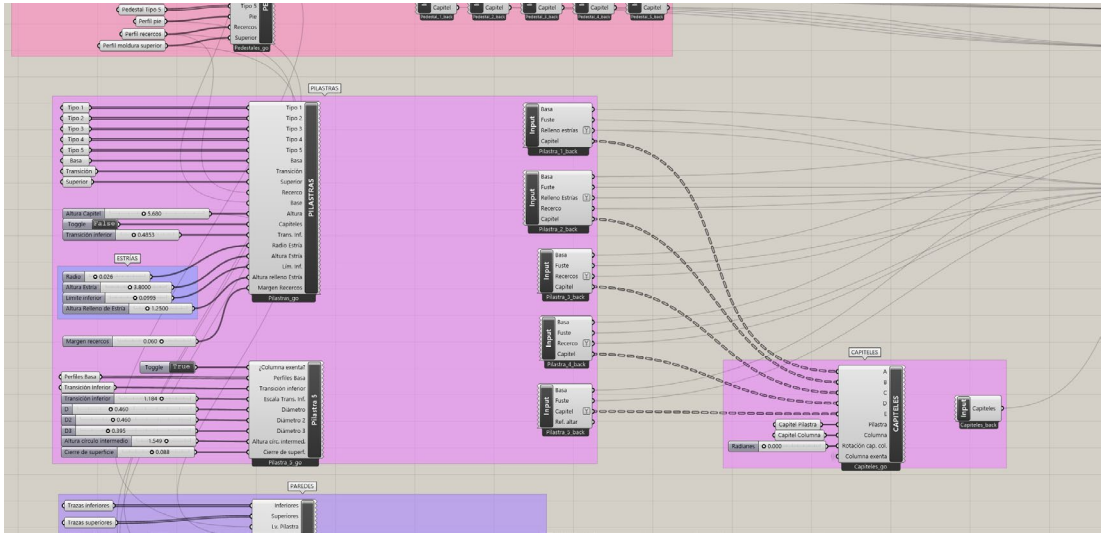

Fig. 6 - Partial view of the main definition (chapel n.3dm)

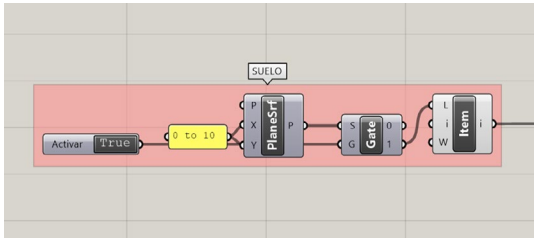

Fig. 7 - Fragment of the main GH definition

DOI: https://doi.org/10.20365/disegnarecon.30.2023.16

ISSN 1828-5961

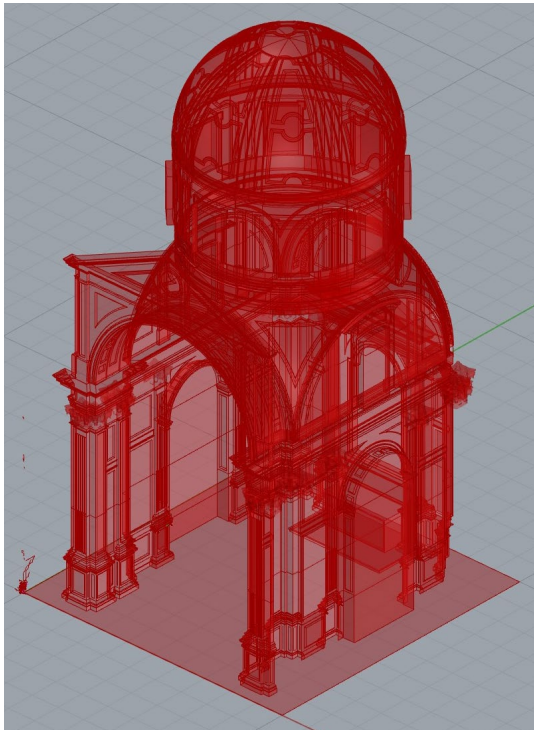

Fig. 8 - Final model before being baked

The definitions of the architectural elements can be related to each other in two fundamentally different ways. The first is in parallel, meaning that the same input parameter affects more than one definition (and is connected to multiple output components). For example, the numeric parameter set to adjust the entablature height also affects the height of the walls, the height from which the pilasters start, etc. The second way of relating is in series, meaning that the results of certain component definitions become the input parameters for other components. This occurs in two specific cases: first, the pendentives are modeled based on the final geometry of the lunettes and the drum, and second, the capitals of the pilasters are arranged based on the information retrieved from their shafts.

Once all the scripts are ready it is possible to model every chapel by adjusting parameters to perfectly adapt the parametric shapes to each chapel's point cloud. Once all the models are perfectly adjusted, the final step is to bake the geometry. This process yields a final 3D model in Rhinoceros that can no longer be modified from Grasshopper. This final model can then be exported to other programs for various purposes (HBIM, rendering, etc.).

3. RESULTS

The main outcome of our research is a new pipeline for parametric 3D architectural modeling, which offers time-saving benefits when modeling similar spaces. With this methodology, we constructed a system of algorithms (Fig. 5) for modeling a collection of similar architectures, referred to as the genotype.

Once the genotype is assembled and thoroughly debugged, ensuring the resolution of any potential errors or bugs that commonly arise during script development, we can effortlessly generate new digital models by providing alternative data configurations. Each new model represents a distinct instance with a different metric configuration, resulting in a unique phenotype. This process can be repeated as needed to obtain all instances (phenotypes) of the same genotype. Notably, in our case study, we successfully modeled the seven neoclassical chapels of the Valencia Cathedral within a matter of minutes using this innovative methodology.

It is worth noting that, as observed in Figure 9, each of the chapels exhibits geometric differences compared to the others. While the metric differences are hardly noticeable to the naked eye due to minimal dimensional variations, there are notable formal differences. Perhaps the most significant variations lie in the dome floor plans [6] and the presence or absence of side altars. It is evident that despite being typologically identical, the chapels differ geometrically from one another. These variations contribute to the unique identity and spatial qualities of each chapel. Although the metric differences may be subtle, they are significant in terms of accurately representing the individuality of each chapel within the larger architectural ensemble. Nevertheless, the proposed system is capable of modeling each and every case of lateral chapels.

The ability of our system to successfully model all the different cases of lateral chapels demonstrates its versatility and adaptability. By incorporating parametric modeling techniques, our methodology effectively captures the underlying architectural principles that define these chapels while accommodating the geometric variations that exist between them. This ensures that the resulting digital models accurately represent the distinct characteristics and spatial configurations of each lateral chapel.

However, it is important to assess the metric accuracy of the obtained models to determine the quality of the results. To achieve this, we convert the final model, consisting of mathematical surfaces, into a polygonal mesh composed of triangles. This mesh is exported in PLY format and imported into software that allows us to handle and compare point clouds with polygonal meshes. In our case, we selected Cloud Compare (Girardeau-Montaut, 2022). We also import the reference point cloud. It is crucial that both models are aligned within the same coordinate system to enable geometric deviation comparison.

Through the analysis of the histogram (Fig. 10), we observed that the majority of points exhibited a deviation of 0.015 m. This level of error is considered acceptable, particularly when considering that the model is constructed using primitive surfaces. The red points on the histogram represent

DOI: https://doi.org/10.20365/disegnarecon.30.2023.16

ISSN 1828-5961

the unmodeled parts, such as the exteriors of the chapels, grilles, statues, and ornaments.

This evaluation allows us to gain insights into the geometric accuracy of the model and provides a measure of its fidelity to the original data. Despite the slight deviations observed, the model captures the essential architectural elements, enabling a comprehensive representation of the neoclassical chapels.

While our methodology has been validated in terms of its metric accuracy, it is important to acknowledge some of its limitations. Our methodology focuses exclusively on the purely architectural elements, such as pilasters, pediments, entablature, and domes, which are geometrically parametrizable. However, it does not encompass the sculptural and ornamental components. The documentation of these elements should follow the standard process of triangulation and optimization of the point cloud, which entails obtaining a sufficiently dense polygonal mesh that allows for optimization through retopology and subdivision surfaces (Panozzo et al., 2011), combined with various texture maps (color, normals, etc.).

It is essential to recognize that by focusing solely on the geometrically parametrizable architectural elements, our methodology may not provide a comprehensive representation of the entire architectural structure. The intricate details of the sculptural and ornamental elements, which contribute significantly to the overall aesthetics and richness of the structure, are omitted. Documenting these elements requires a separate approach, involving the capture of their geometric features through triangulation and the optimization of point cloud data. This typically involves creating a dense polygonal mesh that can be refined using retopology techniques and subdivision surfaces (Fantini, 2012). Furthermore, the inclusion of various texture maps, such as color and normals, enhances the visual fidelity of these elements (Luca Cipriani et al., 2015)

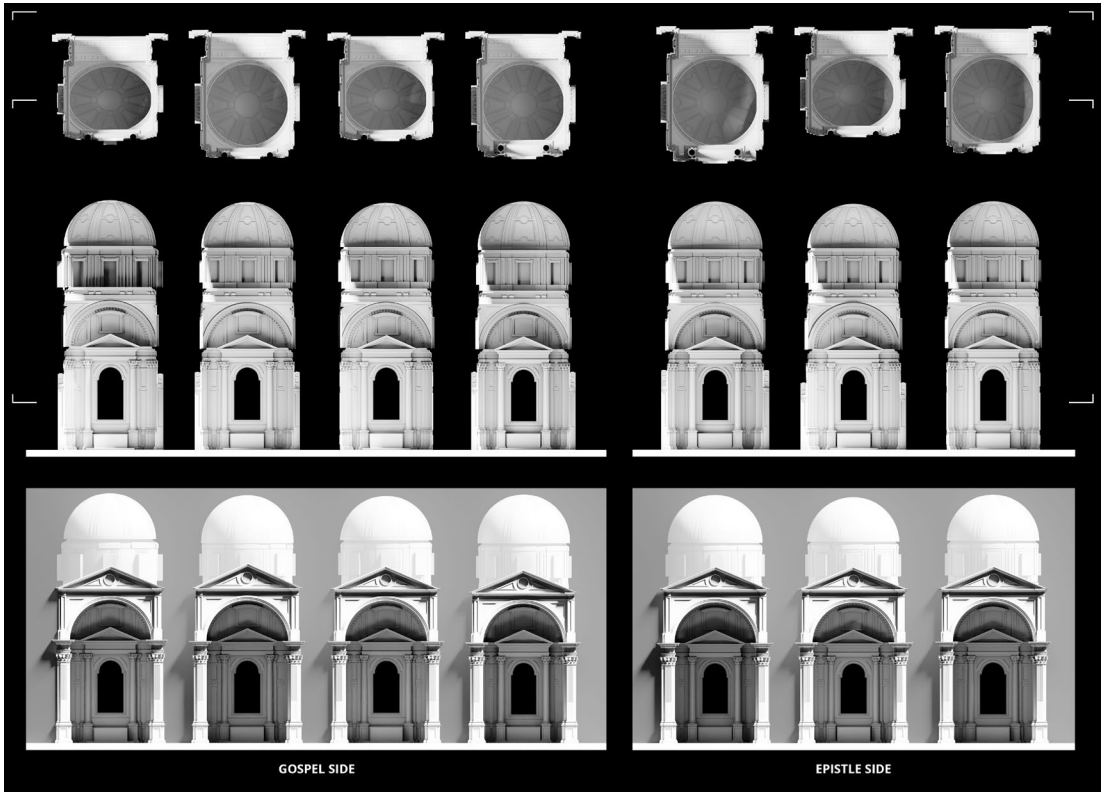

Fig. 9 - Collection of every lateral chapel

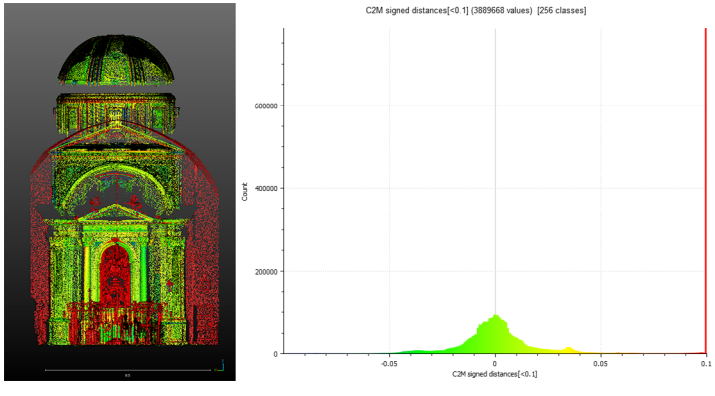

Fig. 10 - Point deviation assesment in comparison with our final model

DOI: https://doi.org/10.20365/disegnarecon.30.2023.16

ISSN 1828-5961

4. CONCLUSIONS

At the threshold of the new era in which artificial intelligence aims to reduce workload in any discipline, it is highly likely that in the near future, research endeavors will emerge to delve into the automation of processes related to architectural surveying. Currently, there are already some works that employ artificial intelligence to semantically segment the meshes generated through digital scanning techniques (Croce et al., 2021) or even attempt to alleviate the workload associated with on-site data collection by recording architectural spaces on video and utilizing AI to generate a digital model—a triangulated mesh along with its corresponding texture (Li et al., 2023). However, these works have not yet provided satisfactory results either by failing to reduce the workload of the operator, as in the first example, or by not meeting the required precision standards in our field of research.

The work carried out in this present research has materialized in a new methodology that indeed reduces the workload in terms of time when modeling multiple instances of similar architectures. As demonstrated, procedural and parametric modeling can be utilized not only for designing projected architecture but also for surveying existing architecture and heritage. The proposed methodology explores novel approaches to 3D-model parameterizable architecture, such as neoclassical architecture, which traditionally follows proportional and modulatory principles derived from classical treatises.

In the context of our research, the point cloud density has proven adequate for faithfully defining the architectural elements. In the event that obtaining sufficient information in terms of mesh density was not feasible due to technical defects or potential volumetric losses in the original architectural elements, our methodology would enable us to turn to architectural treatises to address these gaps. Specifically, in the case of Valencian Neoclassicism, consulting the treatises by Palladio and Vignola would be opportune.

Furthermore, as demonstrated, the methodology ensures high-quality results, and the obtained digital models can be used for various purposes, including HBIM (Jordan-Palomar et al., 2018). In fact, one could argue that they are more suitable for this purpose than the triangulated meshes resulting from the well-known process of point cloud triangulation and optimization (Baviera Llópez et al., 2016), since BIM software natively operates with the same type of geometries and primitive surfaces.

It is true that the parametric modeling of the main architectural elements overlooks the intricate ornaments found in this type of architecture. Such objects necessarily need to be modeled using optimizable polygonal meshes through subdivision surfaces (Fantini, 2012). Therefore, the integration of both types of objects and their inclusion in the HBIM workflow remains a subject for future research endeavors.

Fig. 11 - Mathematical 3D model of a lateral chapel

Fig. 12 - Nadir view of a lateral chapel model

DOI: https://doi.org/10.20365/disegnarecon.30.2023.16

ISSN 1828-5961

NOTE

[1] We are considering the seven neoclassical lateral chapels. There is one extra side chapel -Saint Peter's- whose composition being baroque differs from the others.

[2] The choosen software for this purpose is Geomagic Design X.

[3] The procedures outlined in this study have the potential for application in alternative parametric design software platforms. Although we have abstracted the various phases of our methodology within the pipeline, when elucidating the practical implementation, we find it essential to reference the working paradigm of a specific software. This approach is taken to ensure a lucid description of the operations, as excessive abstraction may result in a level of generality that impedes comprehension of the process.

[4] Grasshopper also allows the incorporation of Python, C#, or RhinoScript code by encapsulating it within components. In fact, many developers have created complementary component palettes for performing specific tasks. These can be downloaded from https://www.food4rhino.com/.

[5] Node-based programming systems are becoming increasingly prevalent in various software applications. They enable the connection of input parameters to actions in a graphical manner, which is more intuitive than writing code. As such, they are suitable for parametric design in various domains, including 3D modeling, texturing, compositing, video game programming, and more. They provide an alternative or complement to application programming interfaces (APIs) offered by certain software programs.

[6] It is possible to parameterize a spherical dome as a special case of an ellipsoidal dome. An ellipsoid is a three-dimensional figure that can have three axes of different lengths. In the case of a sphere, all axes are equal. The general equation for an ellipsoid with three different axes is:

$(x^2 / a^2) + (y^2 / b^2) + (z^2 / c^2) = 1$

For the specific case of a sphere, the equation is:

$(x^2 / r^2) + (y^2 / r^2) + (z^2 / r^2) = 1$

REFERENCES

Baviera Llópez, E. M., Llopis Verdú, J., Denia Ríos, J. L., & Marín Tolosa, R. E. (2016). INTEGRATION OF THREE METHODOLOGIES IN ORDER TO OBTAIN A COMPLETE MODEL OF THE LABORATORIES BUILDING OF FONTILLES . XIII Congresso della Unione Italiana del Disegno, 1-8.

Benedetti, B., Gaiani, M., & Remondino, F. (2010). Modelli digitali 3D in archeologia: il caso di Pompei.

Cipriani, L., García-León, J., & Fantini, F. (2019). FROM the GENERAL DOCUMENTATION of HADRIAN'S VILLA to DESIGN ANALYSIS of COMPLEX CUPOLAS: A PROCEDURAL APPROACH. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, 42(2/W15), 327-334. https://doi.org/10.5194/isprs-archives-XLII-2-W15-327-2019

Cipriani, Luca, Fantini, F., & Bertacchi, S. (2015). El color en las piedras y en los mosaicos de Rávena: nuevas imágenes de los monumentos antiguos a través de la fotogrametría no convencional de última generación. EGA. Revista de expresión gráfica arquitectónica, 20(26), 190. https://doi.org/10.4995/ega.2015.4052

Cortés Meseguer, L. (2014). La construcción del proyecto neoclásico de la catedral de valencia. Universitat Politècnica de Valencia.

Croce, V., Bevilacqua, M. G., Caroti, G., & Piemonte, A. (2021). Connecting geometry and semantics via artificial intelligence: From 3D classification of heritage data to H-BIM representations. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, 43(B2-2021), 145-152. https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-145-2021

Ebert, D., Musgrave, F. K., Peachey, D., Perlin, K., Worley, S., Mark, W. R., & Hart, J. (2002). Texturing and Modeling: A Procedural Approach: Third Edition.

Fantini, F. (2012). MODELOS CON NIVEL DE DETALLE VARIABLE REALIZADOS MEDIANTE UN LEVANTAMIENTO DIGITAL APLICADOS A LA ARQUEOLOGÍA. EGA. Revista de expresión gráfica arquitectónica, 17(19), 306-317. https://doi.org/10.4995/ega.2012.1383

Girardeau-Montaut, D. (2022). Cloud Compare (2.12). https://www.cloudcompare.org/

Jordan-Palomar, I., Tzortzopoulos, P., García-Valldecabres, J., & Pellicer, E. (2018). Protocol to manage heritage-building interventions using heritage building information modelling (HBIM) (Vol. 10, Número 4) [Universitat Politècnica de València. Departamento de Ingeniería de la Construcción y de Proyectos de Ingeniería Civil - Departament d'Enginyeria de la Construcció i de Projectes d'Enginyeria Civil]. https://doi.org/10.3390/su10040908

Li, Z., Müller, T., Evans, A., Taylor, R. H., Unberath, M., Liu, M.-Y., & Lin, C.-H. (2023). Neuralangelo: High-Fidelity Neural Surface Reconstruction. 8456-8465. https://research.nvidia.com/labs/dir/neuralangelo%0A

Molina-Siles, P. (2016). Parametric environment. The handbook of Grasshopper. Nodes & exercises. Parametric environment. The handbook of Grasshopper. Nodes & exercises.

Panozzo, D., Puppo, E., Tarini, M., Pietroni, N., & Cignoni, P. (2011). Automatic construction of quad-based subdivision surfaces using fitmaps. IEEE Transactions on Visualization and Computer Graphics, 17(10), 1510-1520. https://doi.org/10.1109/TVCG.2011.28

Rodrigues, E. (2014). Automated Floor Plan Design: Generation, Simulation, and Optimization (Número 1). UNIVERSITY OF COIMBRA.

Schumacher, P. (2014). Design Parameters to Parametric Design (pp. 1-18). The Routledge Companion for Architecture Design and Practice: Established and Emerging Trends.

Woodbury, R., Williamson, S., & Beesley, P. (2011). Prarmetric Modelling As a Design Representation in Architecture: a Process Account. Proceedings of the Canadian Engineering Education Association (CEEA), July. https://doi.org/10.24908/pceea.v0i0.3827

DOI: https://doi.org/10.20365/disegnarecon.30.2023.16