

## SEMI-AUTOMATIC GENERATION OF AS-BUILT BIM FAÇADE GEOMETRY FROM LASER AND IMAGE DATA

SUBMITTED: July 2013

REVISED: January 2014

PUBLISHED: January 2014 at <http://itcon.org/2014/2>

EDITOR: Ghang Lee

**Conor Dore, PhD Candidate,**

*School of Surveying and Construction Management, Dublin Institute of Technology, Dublin, Ireland;*

*conor.dore@mydit.ie*

**Maurice Murphy, PhD,**

*School of Surveying and Construction Management, Dublin Institute of Technology, Dublin, Ireland;*

*Maurice.Murphy@dit.ie*

**SUMMARY:** *This article outlines a new semi-automatic approach for generating accurate BIM facade models for existing buildings from laser and image data. Two new developments for as-built BIM modelling are presented in this article. The first is a new library of reusable parametric objects designed for modelling classical architectural elements from survey data. These library objects are dynamic and have parameters that can instantly alter the shape, size and other properties of objects. The parametric architectural objects have been designed from historic manuscripts and architectural pattern books. These parametric objects were built using an embedded programming language within the ArchiCAD BIM software called Geometric Description Language (GDL). The second development which is described in more detail in this article is a complete parametric building façade. This parametric building facade can be used as a template for fast and efficient generation of BIM facade geometry. The design of this parametric façade incorporates concepts from procedural modelling which is an automated approach to generating 3D geometries based on rules and algorithms. Parametric architectural objects are automatically combined with rules to generate many different façade arrangements which are controlled by user parameters. When automatically generating a façade, the initial position and size of elements are estimated using classical architectural proportions. Object can then be graphically edited individually or in groups to match the computer generated geometry to survey data. The parametric façade template has also been implemented with the Geometric Description Language for ArchiCAD BIM software. This enables the tools developed to utilise the full benefits of BIM software which includes automated construction or conservation documents, semantic object oriented objects based on IFC semantic classes, automatic lists of objects and material and the ability to add and link additional information to the model. Initial user tests have indicated that the parametric façade is more efficient than previous methods for creating accurate façade models from survey data. The façade template also provides an easier solution for generating façade models when compared to existing methods. Non-specialist users with little experience in 3D modelling can easily generate and modify the façade template by altering parameters graphically or from a dialogue box.*

**KEYWORDS:** *Building Information Modelling, As-Built BIM, Parametric Modelling, Procedural Modelling, Historic Building Information Modelling, Laser Scanning, Photogrammetry*

**REFERENCE:** *Conor Dore, Maurice Murphy (2014). Semi-automatic generation of as-built BIM façade geometry from laser and image data. Journal of Information Technology in Construction (ITcon), Vol. 19, pg. 20-46, <http://www.itcon.org/2014/2>*

**COPYRIGHT:** © 2014 The authors. This is an open access article distributed under the terms of the Creative Commons Attribution 3.0 unported (<http://creativecommons.org/licenses/by/3.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



## 1. INTRODUCTION

The concept of Building Information Modelling (BIM) has received a lot of attention in the architecture, engineering, construction and facility management (AEC/FM) communities due to its ability to enhance communication between the various stakeholders involved in the different stages of a building life cycle (Tang et al 2010). BIM software incorporates an object oriented modelling concept where building elements are represented in 3D with rich semantic objects and attributes containing information about each building element. This enables data relating to a building project to be stored and managed which has great potential for many types of analysis and applications in the AEC/FM communities. There are many advantages of BIM which have led to its recent adoption in the AEC/FM industries. These benefits include improved visualisation, improved productivity due to its ability to easily store and retrieve information, better outcomes through collaboration, greater predictability, reduced waste and whole life cycle management.

BIM models for proposed new buildings are created during the design stages of a project and used throughout all construction stages. BIM is also used for managing buildings throughout their complete lifecycle, including management for renovations or retrofit projects and documenting historical or important buildings. When BIM is used for these applications after a building has been constructed accurate as-built or as-is BIM models are required. Design BIM models or BIM models created from design drawings are not sufficient for managing a building throughout its life cycle as many applications such as retrofit require detailed and accurate models of what actually exists. Design models may not accurately represent what exists as a building may not have been constructed exactly as the design specified (Tang et al 2010). Subsequent changes and renovations to a building over time will also not be recorded on design drawings or models. Recording of historical and significant buildings also require detailed information about the current condition of a building and any damage or deformation to a building. For these reasons the use of BIM for managing buildings after construction stages require the condition of a building to be accurately recorded after a building has been constructed (as-built) or the current condition of a building (as-is).

The concept of an as-built BIM is relatively new and can include both an as-built or as-is representations of a building with BIM. Hichri et al (2013) define an as-built BIM as a building described with a BIM representation at the moment of survey. The creation of an as-built BIM requires accurate measurements to be taken on a building which can be used to create a 3D BIM representation. There are many surveying and photogrammetric techniques available for recording the information needed to create an as-built BIM. Laser scanning technology has become very popular for generating as-built BIM models. This is due to the speed at which it can capture data and the level of detail and accuracy in the resulting point cloud information. Other traditional survey technologies such as total stations and GNSS cannot record the same level of detail at the speed possible with laser scanning. Photogrammetric techniques are also very well suited to capturing data for generating as-built BIM models. Photogrammetric techniques use images taken at different viewpoints to record the 3D geometry of a building or object. Photogrammetric techniques can produce similar results to laser scanning such as point clouds, mesh models and orthographic imagery.

The creation of an as-built BIM can be divided into three main stages. This includes data acquisition, pre-processing of survey data and 3D modelling with BIM. The third stage of modelling with BIM is the longest stage of an as-built project. This is a reverse engineering process where BIM components are created and mapped to the survey data to create the BIM model. Currently there is little or no automation in the modelling stage for generating as-built BIM models. The current practice for creating as-built BIM models is using manual modelling techniques which can be time consuming, labour-intensive, tedious and requiring skilled workers (Tang et al 2010). Current BIM software is also very limited when it comes to dealing with large survey datasets. Most BIM software is currently not capable of handling very large point clouds. As a result other products such as segmented point clouds, cut-sections and orthographic images have to be used within BIM software for the reverse engineering process. Because BIM is mainly used for design of new buildings, commercial software is currently limited with tools for modelling non-ideal geometries that often occur in existing buildings due to deformation and damages over time. For the same reason modelling tools are also limited when it comes to creating complex organic and non-uniform architectural elements that are often found in old and historical buildings. The time consuming and tedious nature of the as-built BIM process motivates the need for new more automated solutions designed specifically for modelling existing buildings from survey data. Many automated solutions for modelling are being developed in other areas such as computer vision, gaming and film production.

Some of these techniques as outlined by Tang et al (2010) are very promising but have not to date been applied in the AEC/FM industries.

This research aims to address these current issues by providing new tools for creating accurate as-built BIM models from laser and image data. These developments are packaged as plug-ins for existing BIM software and contain two main elements. The first is a new library of interactive parametric elements designed for modelling classical architectural elements found in many existing and historical buildings. These library objects are dynamic and have parameters that can instantly alter the shape, size and other properties of objects. Included with this library is a process for mapping objects to survey data. Design of this library is summarised in this article. Previous work such as Murphy et al (2013) describes the creation of this library in more detail.

The second development which is described in detail in this article is a complete parametric façade which can be used as a template for modelling endless configurations of building façades with a semi-automatic procedure. This parametric façade template is also being developed as a plug-in for existing BIM software. The design of this parametric façade incorporates concepts from procedural modelling which is an automated approach to generating 3D geometries based on rules and algorithms.

## **1.1 Overview of Article**

This article firstly contains a review of existing work related to this research. Next the design of a new library of parametric architectural objects is described. This library is called Historic Building Information Modelling (HBIM). Section 4 then describes the design and implementation for the parametric façade which can be used to semi-automatically model existing building facades from survey data. Section 5 describes the procedure for altering the parametric façade in order to accurately model façades from survey data. Section 6 contains a case study as an example of models created with the developed library objects and parametric façade. Section 7 describes the initial testing and evaluation of the parametric façade template and finally sections 8 and 9 contain a conclusion and future work to be carried out.

## **2. RELATED WORK**

Previous research into as-built BIM modelling has concentrated on reviewing the current manual processes and workflows. To date little or no automation has been introduced for this long and tedious task. Hichri et al (2013) and Tang et al (2010) provide a very good overview of the current workflows for as-built BIM modelling along with an overview of automated techniques used in other areas but not yet implemented for BIM modelling for AEC/FM communities.

Another area where as-built BIM modelling is being adopted is in the cultural heritage field. The use of BIM software in the cultural heritage field has many advantages such as semantic object oriented modelling which allows for the classification of heritage objects, automatic lists of objects and material and automated conservation documents. BIM allows for accurate modelling using interactive parametric objects that contain information about the objects and their relationships with other objects. One of the current limitations of BIM in the cultural heritage field is the lack of parametric library objects within BIM software that could be used for historical buildings or heritage sites (Boeykens 2011). The HBIM plug-in presented in this article intends to fill this current gap and contains a library of parametric objects that can be used to accurately model classical style architecture for heritage applications.

Work by Boeykens et al (2012), Fai et al (2011) and Pauwels et al (2008) have concentrated on as-built modelling of heritage sites using BIM software. Pauwels et al (2008) describe an approach called Architectural Information Modelling which uses BIM software to document a heritage site which contains geometric data along with appended historical information such as photographs, scanned documents or research material. Fai et al (2011) adopt a similar approach which links heritage information to a BIM model but also includes documentation related to tangible and intangible heritage. Boeykens et al (2012) use BIM software to create a reconstruction of the Vinohrady Synagogue in Prague which was demolished in 1951. Existing tools in ArchiCAD's BIM software such as the Shell tool are used to try and create organic geometry that is needed for the heritage site. The authors of this paper note that the BIM software used was almost completely focussed on contemporary buildings and that more specific tools are needed for historical reconstructions.

Very little work has been carried out in the development of parametric historical objects that can be used for heritage applications. Work by Chevrier et al (2010) is one of the few examples of parametric modelling for historical buildings. In this work a library of parametric objects is created using a MEL scripting language for Maya software from Autodesk. These models are not suitable for BIM software though and their primary application is documentation and visualisation.

Another approach adopted by Thaller et al (2011) includes the use parametric and procedural modelling techniques to create eight parametric building models. These parametric building models are designed as a planning tool to analyse buildings as they change from design stage to construction stages. Their approach involved using a scripting language for parametric design called Generative Modelling Language (GML). Procedural modelling techniques using shape grammars, derived from formal grammars are applied with the GML scripting language to create the parametric building models. The resulting building models can be manipulated interactively with respect to dimensions, number of floors and other parameters. These models are not designed for a detailed representation of existing buildings but as a rough model to analyse the impact of changes in the building on the energy footprint. Although these models are not detailed enough for as-built modelling and not implemented for BIM software, similar techniques could be adopted as a framework for generating as-built BIM models for the AEC/FM communities.

Automatic methods for generating geometric models such as procedural modelling have traditionally been used in applications such as film and gaming where content can be randomly generated based on rules and algorithms. The use of shape grammars in procedural modelling has gained a lot of interest and is now being used to generate content for architectural modelling. Shape grammars originally introduced by Stiny and Gips (1972) are derived from formal grammars and consist of a set of basic vocabulary shapes (terminals and non-terminals) and a set of production rules to transform these shapes to create 3D content. A shape grammar called CGA Shape (Muller et al 2006) has been developed for the commercial software CityEngine from ESRI. This shape grammar is designed for procedural modelling of buildings and cities. CityEngine software provides users with tools to create 3D content from scripts using this shape grammar. With this software it is possible to procedurally generate buildings from 2D footprints and other GIS datasets for modelling existing buildings and cities. While these models contain information about semantics and can be automatically generated, they lack the detail that would be required for applications in the AEC/FM communities.

Most procedural modelling applications require users to code rules in a grammar to create a specific model. This text based approach restricts users with little computer science background. Work by Lipp et al (2008) has concentrated on creating an interactive visual editing tool for shape grammars to create rules from scratch without the need for text file editing. This makes automated approaches for generating 3D content much more accessible and does not require advanced users to create scripts to code rules.

Another approach (Müller et al 2007) uses CGA shape grammar for automatic modelling of building facades from a single rectified image. This method automatically detects the façade structure using mutual information and symmetry detection to divide the façade into floors and tiles. Further tile refinement is automatically carried out using edge detection to split tiles into smaller regions using a subdivision concept from procedural modelling which creates a hierarchy of elements. This is used to automatically detect window positions, ledges and window sills. Finally 3D objects from a library of architectural elements are matched to the subdivided façade to add windows and other architectural elements. Depth for different sections of the façade is added manually and the computed façade can be exported as shape grammar rules in the CGA Shape Grammar. This method shows how procedural modelling techniques can be applied to existing buildings. This method works well for urban environments where facades contain a lot of repetition and symmetry can be easily detected. However less repetitive facades with a lot of architectural detail may be problematic for this automatic method.

Other work by Hohmann et al (2009) use shape grammars to automatically model building facades for automatic 3D city reconstruction. This project called CityFit aims to reconstruct 80% of the buildings in the city of Graz fully automatically. Their workflow uses road side photographs and LIDAR point clouds as input data. Image based feature detection is carried out using a partial shape fragment matcher to detect and segment windows, arches and other decorative elements. Depth maps derived from the point cloud are also used to provide depth information. The results from these segmentations are matched against a set of shape grammar templates obtained by façade analysis and classification. The use of shape grammar templates in the modelling stage incorporates architectural knowledge to automate this stage. The shape grammars and shape grammar templates

used are based on the concepts of the CGA shape grammar but have been implemented using the Generative Modelling Language (GML).

Approaches such as procedural modelling with shape grammars have many advantages such as automatic generation, great flexibility for variation, object hierarchy, scalable geometric representation and data handling of large models. However a big disadvantage of these methods for existing buildings is that they lack the detail and accuracy required for most as-built applications in construction and heritage fields. Muller et al (2006) state that manual methods are used instead of procedural modelling techniques to create detailed elements such as roof bricks, capitals and window grills. Procedural modelling also does not automate the production of engineering drawings for construction or conservation documentation. The accuracy of automatic techniques such as edge detection from images and object recognition is also not suitable for producing accurate models needed for construction documentation. Applications for procedural modelling of existing facades (Müller et al 2007) are primarily visualisation and the aim is to automatically create a geometric model that “looks like a plausible interpretation of the input image” (Müller et al 2007). For as-built documentation a more precise and accurate model is required.

The new approach described in this article differs from existing work (Chevrier et al 2010; Müller et al 2007; Thaller et al 2011) as it combines procedural modelling techniques with detailed parametric modelling. Combining procedural modelling techniques with parametric modelling makes use of the advantages of automated procedural modelling combined with very detailed parametric objects that can be used to model complex geometric detail such as capitals and decorative architectural elements. Using a semi-automatic approach for modelling existing buildings also allows for very accurate models to be created from laser scan or image data. Implementing these techniques with the Geometric Description Language (GDL) allows for automated techniques to be used within a BIM environment making it suitable for applications in the AEC/FM and cultural heritage communities. This introduces automation to the as-built BIM process which is currently a long and tedious manual task.

### **3. PARAMETRIC LIBRARIES – HBIM**

This section describes a summary of the methodology for creating the library of interactive parametric architectural objects that are mapped to laser scan and image data. This library called Historic Building Information Modelling (HBIM) is described in more detail in previous work such as Murphy et al (2013).

#### **3.1 Historic Data**

The design and detail for the parametric objects are based on architectural manuscripts ranging from Vitruvius to Palladio to the architectural pattern books of the eighteenth century. The architecture of the renaissance introduced and documented advanced scientific rules for the production of architectural elements, which support the design of parametric models. The use of historic data introduces the opportunity to develop detail behind the object’s surface concerning its methods of construction and material makeup.

#### **3.2 Geometric Description Language**

The parametric library objects were created using a programming language called Geometric Description Language (GDL), a language for creating parametric objects within ArchiCAD BIM software. This syntax of this language is similar to that of BASIC programming language. GDL provides a large number of functions for creating 3D parametric objects using primitive shapes such as blocks, spheres, cones and ellipses or by generating shapes from 2D outlines. GDL uses coordinate transformation commands stored in a stack to position multiple objects relative to each other. GDL allows for graphical editing of parameters, complex Boolean operations, various control statements and the use of mathematical functions in creating parametric objects. Also provided is the ability to script a specific user interface for objects and their parameters (Graphisoft 2011).

#### **3.3 Shape Rules and Parametric Design**

Using GDL the classical elements detailed in the pattern books are re-produced using a design framework based on parametric rules and shape rules. Objects are scripted with parameters making them dynamic objects that can be reused. A bottom-up approach is adopted which starts with the smallest building objects such as ornamental

mouldings and profiles. These uniform objects are created from a shape vocabulary of 2D shapes that allow for all configurations of the classical orders (FIG. 1). Shape rules are used to transform these 2D shapes to represent classical mouldings and profiles. Non-uniform and organic shapes are developed in GDL through a series of procedures attempting to maximise parametric content of the objects (FIG. 1). These shapes are stored as individual parametric objects or combined to make larger objects such as columns, pediments, walls, windows or roofs.

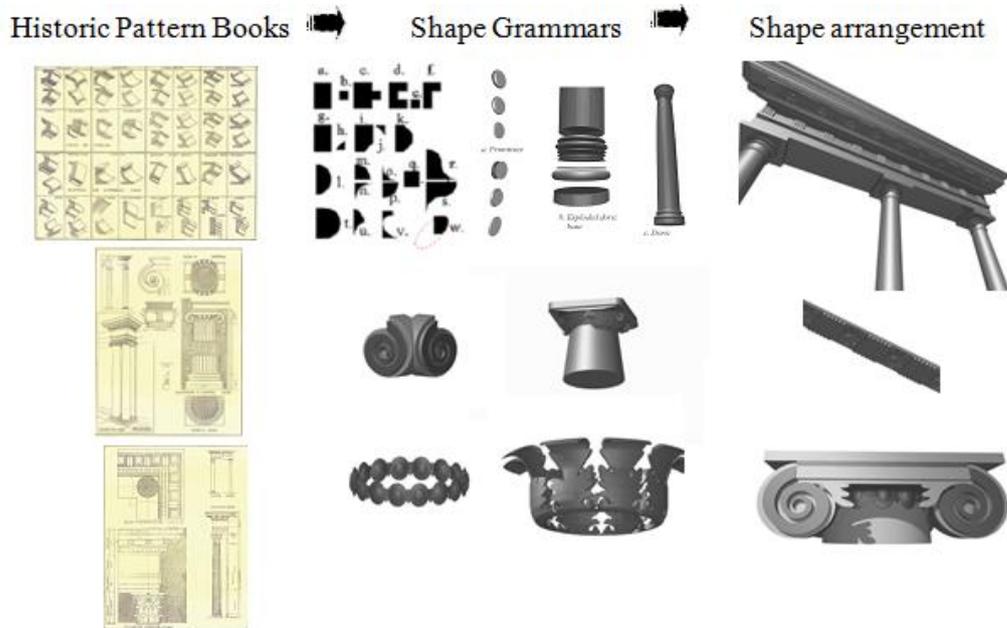


FIG. 1 Sample historic data, shape grammars and shape arrangements for HBIM library objects.

#### 4. PROCEDURAL FAÇADE GENERATION

A parametric building façade has been developed to automatically combine parametric library objects to create various different building façade arrangements. Different façade arrangements are generated by altering parameters such as the number of storeys, number or horizontal tiles and door position. When automatically generating a façade, the initial position and size of elements are estimated using classical architectural proportions. Object can then be graphically edited individually or in groups to match the computer generated geometry to survey data. This section describes the design and implementation stages for developing this parametric façade as a plug-in for the ArchiCAD BIM software.

##### 4.1 Architectural Rules

Similar to the design of the library objects the parametric façade also uses architectural knowledge to assist with the digital reconstruction process. Architectural rules and proportions outlined in pattern books relating to classical building façades are used. The proportioning of a classical façade is determined by the geometry of the window openings, which is expressed in FIG. 2 by the relationship between circles of the same radius. The top windows are made up of a single circle, in the next set of windows intersecting circles and finally in the lower set of windows the circles are placed one on top of each other. Using a parameter for the window width or circle diameter the height and position of windows can be calculated with these proportions. These proportions are evident in most classical buildings, however alterations to buildings can obscure or remove some of the original façade proportions. Alterations to a façade over time can include removal or enlarging of brick walls, window and door openings and parapets. After testing the classical proportions on a variety of surveyed facades (FIG. 3) the most suitable proportions and parameters were adopted as seen in FIG. 4.

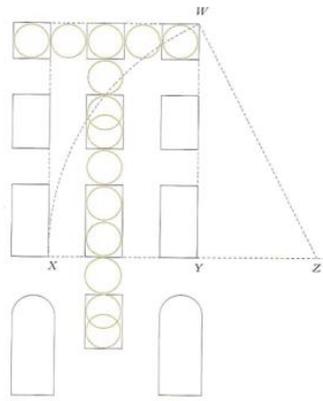


FIG. 2: Proportions for façades and openings.



FIG. 3: Testing classical proportions using orthographic images from surveyed classical buildings.

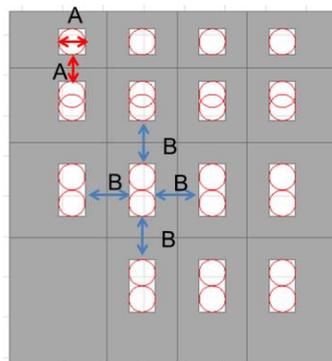


FIG. 4: Proportions and parameters used for design of parametric façade.

## 4.2 Design and Conceptual Framework

The design and conceptual framework for the parametric façade is based on concepts from shape grammars. A shape grammar is a production system used in procedural modelling to automatically generate two or three dimensional geometries from a basic vocabulary of shapes and a set of production rules. A design and conceptual framework based on parametric design and shape grammars is described in this section.

### 4.2.1 Parametric Design

The first stage of design involved evaluating a set of key parameters that would allow the façade template to be modified to model many different building arrangements. Efficient methods for editing these parameters also had to be established. The standard method for editing parameters of a parametric object is to specify or edit parameters from a list in a dialogue box. This approach is adopted for modifying global parameters or parameters affecting the entire façade. Editing more specific parameters relating to an individual object such as a window opening would be very time consuming and inefficient using this method as it would require a large number of parameters to be measured and entered into a dialogue box. To facilitate efficient parameter editing another method using graphical parameter editing has been designed. This method allows users to select a specific part of the façade in 2D or 3D and interactively edit the objects parameters by moving the object graphically. This enables parameters of the façade to be modified while overlaying the model with survey data in 3D or 2D. This removes the need for taking measurements and entering measurements into a dialogue box, instead parameters can be matched to survey data directly in 3D or 2D. FIG. 5 shows an example of the dialogue box for editing parameters from a list (left) and also graphical editing where a parameter for the distance between two floors is being altered in 3D (right).

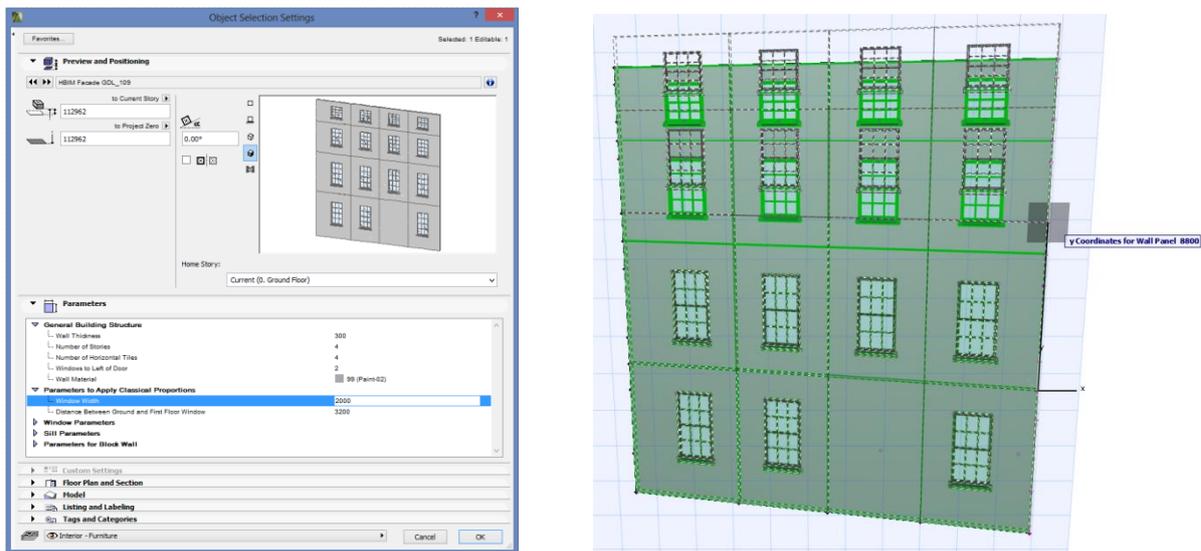


FIG. 5: Modifying parameters from a dialogue box (left) and modifying parameters graphically (right).

The key parameters evaluated for modifying the façade template are show in TABLE 1, along with the parameter type and methods of editing. The first group of parameters are designed to procedurally generate the structure of a façade. Parameters for the “Number of Storeys”, “Number of Horizontal Tiles”, “Door Position” and “Wall Thickness” are included in this group. These parameters can be edited graphically or from a dialogue box. The second group of parameters are designed to calculate and apply classical architectural proportions outlined in section 4.1. These parameters apply a global update to the entire façade and must be entered from the dialogue box. The third group of parameters are designed for graphical editing only and enable any specific object or element on the façade to be altered. This includes graphically editing window corners, distance between floors, distance between vertical columns and façade width and height. Further architectural elements such as windows have additional parameters which can be altered from the dialogue box or by graphically editing specific instances or groups of instances. Certain parameters in TABLE 1 can also be edited in groups for fast editing. This includes editing the height of all windows on a floor simultaneously or the width of all windows in a vertical column simultaneously.

TABLE 1: Key parameters designed for modifying the façade template.

PARAMETER	EDITING METHOD	PARAMETER TYPE
<b>General Structure of Façade:</b>		
Number of Storeys	Dialogue Box & Graphical	Integer
Number of Horizontal Tiles	Dialogue Box & Graphical	Integer
Door Position (windows to left of door)	Dialogue Box & Graphical	Integer
Wall Thickness	Dialogue Box & Graphical	Real Number
<b>Parameters to Calculate Classical Proportions:</b>		
Window Width (Global)	Dialogue Box	Real Number
Distance Between Ground and First Floor Window Openings	Dialogue Box	Real Number
<b>Graphical Modifications Only:</b>		
Distance Between Floors (Tile Coordinates)	Graphical	Real Number (Array)
Distance Between Vertical Columns (Tile Coordinates)	Graphical	Real Number (Array)
Building Width/Height (Tile Coordinates)	Graphical	Real Number (Array)
X,Y Coordinates of Individual Window Opening Corners	Graphical	Real Number (Array)

#### 4.2.2 Shape Grammar

Concepts from shape grammars have been adopted to design and implement the parametric façade as a template for modelling building façades. Shape grammars are a very suitable approach to architectural modelling as they allow models to be created from a vocabulary of basic shapes and set of replacement rules where a shape can be replaced or altered by transformations. These principles facilitate the encoding of classical architectural rules and proportions which also use grammars or vocabularies for architectural elements and rules to combine these basic shapes.

A standard shape grammar introduced by Stiny et al (1972) can be defined as  $\{N, \Sigma, P, S\}$  where  $N$  and  $\Sigma$  represent a finite set of nonterminal and terminal shapes (the vocabulary),  $P$  are a set of production rules and  $S$  is a starting seed. Terminal shapes are the basic vocabulary elements and can be a collection of points, lines, planes areas or solids. Nonterminal shapes are markers or boxes that are used to guide the terminal shapes during generation process and control the scope and position of shapes. The production process begins with nonterminal shapes which are replaced by terminal shapes when rules are applied. The production process terminates when no more rules can be applied and all nonterminal shapes have been removed. Production rules are applied in the form  $A \rightarrow B$  where  $A$  and  $B$  are terminal and nonterminal shapes. When a rule is applied the shape on the left hand side is replaced by a new shape on the right hand side of the rule.

Stiny (1977) also introduced the concept of a parametric shape grammar. This type of shape grammar differs from standard shape grammars in that it contains shape rules defined in terms of parameterised shapes. A parametric shape grammar can be defined as  $\{S, L, R, I, T\}$  where  $S$  is a finite set of shapes,  $L$  is a finite set of labelled points,  $R$  is a finite set of shape rules,  $I$  is an initial shape and  $T$  is a set of transformations. The main difference to the standard shape grammar is that nonterminal shapes are replaced by labelled points or parameters that are associated with shapes. Euclidean transformations have also been added which can be a translation, rotation, scale or mirror. Shape rules are applied to a shape with an assignment of real values to the parameters and with additional transformations as required.

The main concepts from shape grammars that have been adopted for developing the parametric façade for BIM software are the use of a basic shape vocabulary and shape rules to transform the shape vocabularies to automatically generate parametric façade geometry. The design of the parametric façade template using these

shape grammar concepts is described in sections 4.2.3 below. The implementation of these shape grammar techniques with the Geometric Description Language (GDL) is described in Section 4.3.

#### 4.2.3 Design of Parametric Shape Grammar Elements

The shape grammar techniques adopted for the design of the parametric façade incorporate the five elements of Stiny’s Parametric Shape Grammar {S, L, R, I, T} (1977). The initial shape {I} is made up of a labelled solid shape as seen in FIG. 6. This shape contains four labelled points as parameters which control the size and shape of the object. An additional parameter also defines a thickness which converts the 2D shape into a 3D solid shape.

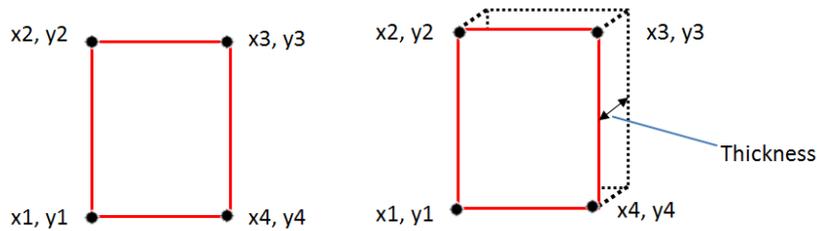


FIG. 6: Initial shape {I} for parametric shape grammar design.

The basic elements that make up the vocabulary of shapes {S} can be seen in FIG. 7. Shapes include two wall tiles; one wall tile *TW* which contains a window opening and surrounding wall. A second wall tile *TD* is used as a panel containing a door opening. Additional library objects relating to a door and door case such as columns and pediments are linked with this shape *TD*. Other shapes include library objects for windows and a simple block that is used to create detail for ashlar stone wall geometry. Additional library objects described in the previous sections are also used in conjunction with these basic shapes in FIG. 7.

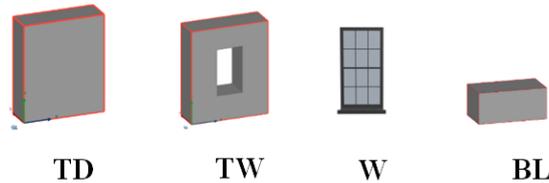


FIG. 7: Basic shape vocabulary elements {S} for parametric shape grammar design.

The shape rules {R} used are shown in TABLE 2. The concepts for many of these rules are similar to the CGA shape grammar, (Muller et al, 2006). Additional custom rules have also been developed for incorporating the architectural rules described in Section 4.1. TABLE 2 shows the initial shape on the left hand side of each rule and the resulting shape on the right hand side after the rule has been applied. Each rule is applied with an assignment of real values to parameters for shapes and transformations if required such as translations, rotations, scaling or mirroring.

Rule 8 is the main rule that creates the façade structure. This rule repeats the wall tile *TW* in both x and y directions based on various parameter settings and architectural rules. This method contrasts to the concepts adopted in the CGA shape grammar (Muller et al, 2006) where a façade is split into smaller tiles. This rule repeats the input shape in the y direction for the first column, and then moves to the second column and so on until all tiles have been placed for the specified parameters (FIG. 8). Parameters for the “number of floors” and “number of columns” control the number of repeated instances to be placed in each column and floor. The input for this rule is the shape *TW* and coordinates for this shape which is an assignment of the shapes parameters.

TABLE 2: Shape rules {R} for parametric shape grammar design.

Rule	LHS		RHS	Description
Rule 1		→		Shape Replacement
Rule 2		→		Repeat x
Rule 3		→		Repeat y
Rule 4		→		Split x
Rule 5		→		Split y
Rule 6		→		Split x & Remove
Rule 7		→		Split y & Remove
Rule 8		→		Conditional Repeat x, y
Rule 9		→		Shape Replacement
Rule 10		→		Shape Replacement

FIG. 9 shows the order and application of the rules shown in TABLE 2 to create the basic parametric façade. The application of rule 9 adds a door tile TD. The position of this is obtained from a user defined parameter “windows to left of door”. The application of rule 10 adds window objects to all window openings on the façade. Global parameters for window objects can be entered and set from the objects dialogue box. Specific parameters for a particular instance of a window object can be set using graphical parameter editing.

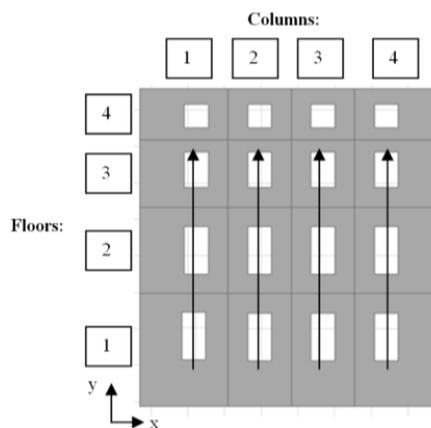


FIG. 8: Repeated wall tile resulting from application of rule 8 to shape TW.

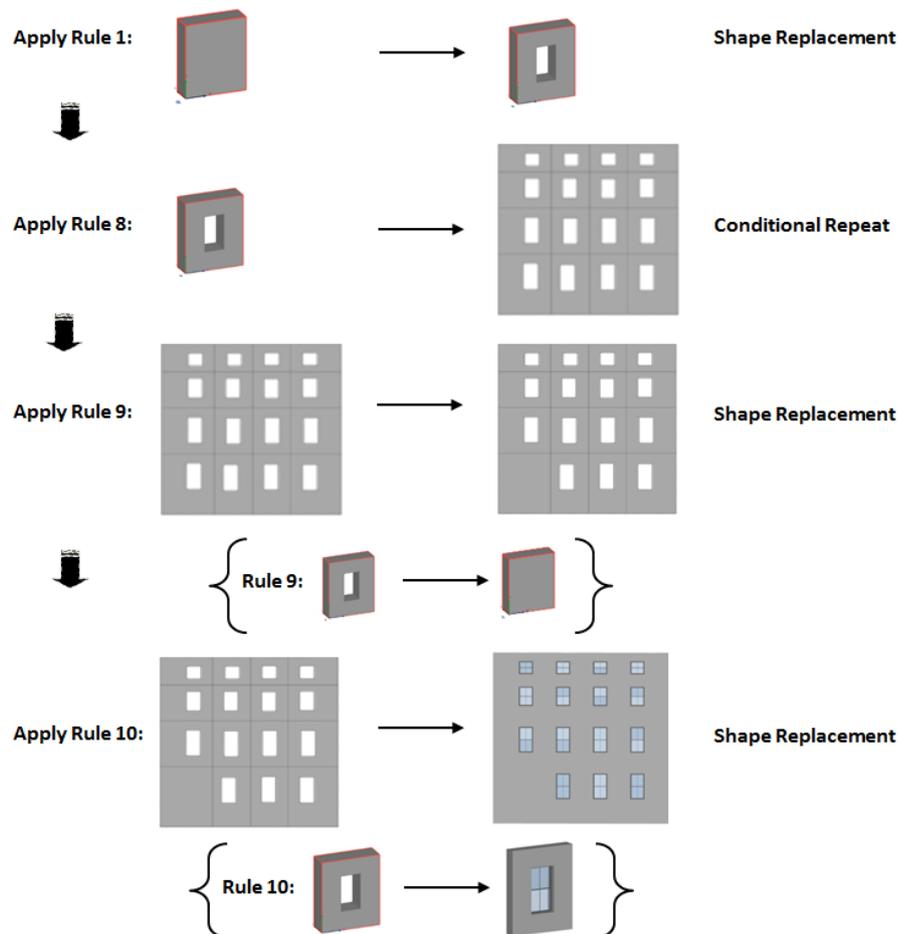


FIG. 9: Application of shape rules specified in TABLE 2.

### 4.3 Implementation with GDL

Various programs and languages have been used to implement shape grammars to generate shapes from a grammar. Examples include the CGA Shape grammar implemented in ESRI's CityEngine software (Muller et al, 2006). Other shape grammars have been implemented using the Generative Modelling Language (GML) (Hohmann et al, 2009) (Thaller et al, 2011). The concepts of a parametric shape grammar described in the previous section have been implemented using the Geometric Description Language (GDL). Although designed for parametric modelling, because of its powerful capabilities GDL can also be used to generate shapes using shape rules and shape vocabularies similar to a shape grammar. To date this language has not been used to implement shape grammars techniques. Using the GDL language with shape grammars techniques enables the advantages of automated procedural modelling to be incorporated within a BIM environment.

Coding with the Geometric Description Language (GDL) is carried out using different script for different parts of the parametric object being created. The 3D script is the main script used to build the parametric 3D object. Other scripts include a 2D script, master script, parameter script, property script and a user interface script. The main shape grammar vocabulary objects and rules developed for the parametric façade have been coded in the 3D script. The 3D script is structured with subroutines and an executive script which is a controlling script used to call subroutines. Shape vocabulary object are stored in individual subroutines and shape rules which are applied to shape vocabulary objects are coded in the executive script. Sample code for shape vocabulary objects can be seen in FIG. 10 and FIG. 11.

```

1  ! =====
2  100:!! WINDOW TILE
3  ! =====
4  GROUP "tw" !!! Window Tile
5
6  cPRISM_ "wmat", "wmat", "wmat", 5+5, thk,
7
8      tx[i],   ty[i],   15,
9      tx[i+1], ty[i+1], 15,
10     tx[i+2], ty[i+2], 15,
11     tx[i+3], ty[i+3], 15,
12     tx[i],   ty[i],   -1,
13
14     wx[i],   wy[i],   15,
15     wx[i+1], wy[i+1], 15,
16     wx[i+2], wy[i+2], 15,
17     wx[i+3], wy[i+3], 15,
18     wx[i],   wy[i],   -1
19
20 ENDGROUP
21 RETURN

```

Coordinates of tile as parameters

Coordinates of window opening as parameters

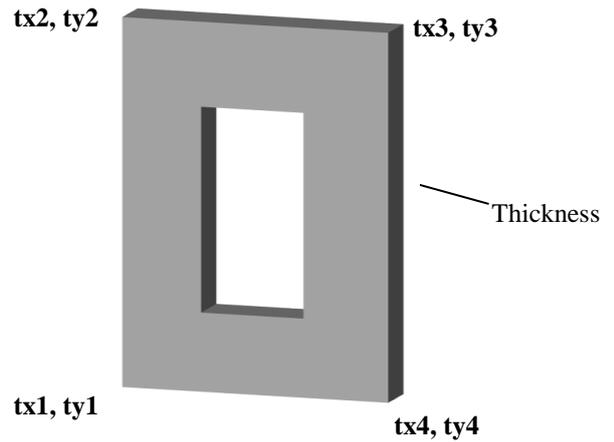
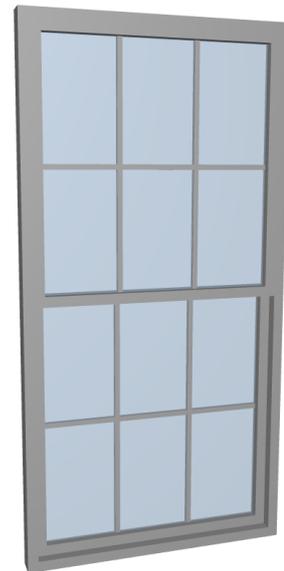


FIG. 10:GDL code for vocabulary shape TW.

```

1  ! =====
2  400:!! SASH WINDOW:
3  ! =====
4  ! =====
5
6  ! =====
7  ! Executive Script for Sash Window:
8  ! =====
9
10 MATERIAL frame_mat
11 PEN win_pen
12
13 GOSUB 410: !!! Window Frame
14 ADD frame_width, frame_width, 0
15 ADDy lower_sash_opening[i]
16 GOSUB 420: !!! Lower Sash Frame
17 DEL 1
18
19 ADD 0, lower_sash_box_height[i]-sash_frame_width, sash_frame_thk
20 ADDy -upper_sash_opening[i]
21 GOSUB 430: !!! Upper Sash Frame
22 DEL 3
23
24 ! =====
25 !!! Lower Glazing Bars
26 ! =====
27
28 ADD frame_width+sash_frame_width, frame_width+sash_frame_width, 0
29 ADDy lower_sash_opening[i]
30
31 FOR iii=1 TO (hor_lower[i]-1)
32 ADDx lgx[i][iii]
33 GOSUB 470: !!! Vertical Glazing Bar Lower
34 DEL 1
35 NEXT iii

```



W

FIG. 11: Sample script for parametric sash window as a vocabulary shape in a subroutine.

The shape rules described in section 4.2.3 are coded with GDL in an executive script located at the top of the 3D script. When a rule is applied to a shape vocabulary object the required object is called from its subroutine and altered by the rule. Rules two and three from TABLE 2 are implemented in GDL using loop commands to repeat shapes in a certain direction. The GDL functions CUTPLANE and CUTPOLY are used for rules four and five to split geometry into multiple components. Rules six and seven use Boolean operations in GDL such as SUBTRACT to remove segments after splitting. Rule eight, the main rule used to generate the structure of a façade is implemented with loop commands to repeat the input shape TW in both x and y directions. Arrays are used during this loop to store unique parameter values for each iteration of the repeated object. While repeating

the shape *TW*, IF statements are used to control the number of repeated instances in a particular column and also the number of columns to be placed. The user parameters for “Number of Floors” and “Number of Horizontal Tiles” are used to control this. Coordinate transformation are also used with this loop command to control where the next instance is to be placed (FIG. 12).

Rule nine from TABLE 2 is used to place a door tile (*TD*) into the generated facade. The position of a door tile is obtained from the user defined parameter “Windows to left of Door”. Rule ten from TABLE 2 is used to add the vocabulary shape for a window object (*W*) to existing window tiles (*TW*). The width and height of each repeated window object is automatically calculated from the size of the window openings in each window tile.

```

1  ! -----
2  !!! IF STATEMENTS TO MOVE ORIGIN DURING LOOP:
3  ! -----
4
5  IF i=5 OR i=21 OR i=37 OR i=53 OR i=69 OR i=85 OR i=101 OR i=117 THEN
6      ADDy py[2]
7  ENDF
8
9  IF i=9 OR i=25 OR i=41 OR i=57 OR i=73 OR i=89 OR i=105 OR i=121 THEN
10     ADDy py[6]
11  ENDF
12
13 IF i=13 AND H=2 OR i=29 AND H=2 OR i=45 AND H=2 OR i=61 AND H=2 OR i=77 AND H=2 OR i=93 AND H=2 OR i=109 AND H=2 OR i=125 AND H=2 THEN
14     ADDy py[2]
15  ENDF
16
17 IF i=13 AND H=3 OR i=29 AND H=3 OR i=45 AND H=3 OR i=61 AND H=3 OR i=77 AND H=3 OR i=93 AND H=3 OR i=109 AND H=3 OR i=125 AND H=3 THEN
18     ADDy py[6]
19  ENDF
20
21 IF i=13 AND H=4 OR i=29 AND H=4 OR i=45 AND H=4 OR i=61 AND H=4 OR i=77 AND H=4 OR i=93 AND H=4 OR i=109 AND H=4 OR i=125 AND H=4 THEN
22     ADDy py[10]
23  ENDF
24
25 FOR oi=17 TO 113 STEP 16
26     IF i=oi AND H=2 THEN
27         ADDx px[i-13]
28         ADDy -py[i-15]

```

FIG. 12: Sample code showing coordinate transformations embedded in a loop as part of rule eight (TABLE 2).

A master script has also been used in implementing the parametric façade with GDL. A master script has been used to set up parameters, populate parameter arrays, define global parametric expressions and define material for objects. The classical proportions outlined in section 4.1 are set up and coded in the master script. These proportions provide an initial estimate for the position and size of elements on the façade which can be later edited to accurately match survey data. These proportions are calculated from two user defined parameters for “window width” and “distance between ground and first floor windows”. A set of parametric expressions calculate the classical proportions from the two user defined parameters. Sample GDL code for these parametric expressions can be seen in FIG. 13. Another feature of the developed parametric façade is the ability to graphically edit groups of parameters simultaneously. This allows for very quick editing of elements on the façade. For example it is possible to edit the height of all windows on a floor simultaneously along with separate editing of individual window heights. The coding to allow for editing groups of objects simultaneously is also implemented in the master script. An example of various façade models automatically generated by altering parameters for the developed parametric façade can be seen in FIG. 14.

#### 4.3.1 Non-Uniform and Irregular Geometries

One of the main challenges involved with modelling existing buildings is accurately representing the current condition of a building which may include damage or deformation to a building or building parts caused over time. Damage caused by environmental conditions, settlement of the building on the ground or other causes may result in irregular or non-uniform geometries such as walls not intersecting exactly at 90 degree angles or non-vertical walls. Creating an accurate ‘as-is’ representation of an existing building should include these irregularities to ensure an accurate representation of the building. Most architectural modelling software (including BIM software) is focused on new building designs and as a result has very limited tools for modelling irregular and non-uniform building elements. In order to accurately model existing buildings the new parametric

façade was developed with options to create non-uniform and irregular geometries. This includes parameters to specify the inclination angle for non-vertical walls, options to specify exact corners for window openings which do not have to be orthogonal. Windows automatically placed in these window openings are adjusted with irregular window frames to fit irregular wall openings. Newly developed parametric sash window objects also include options for specifying irregular grid positions for glazing bars. Future work will involve developing further options for irregular and non-uniform geometry often found in existing buildings. The next section outlines how this parametric façade template can be used to create accurate façade models from survey data.

```

344 !!! x-coordinates for 1st set of vertical panels:
345
346 FOR i=1 TO 16 STEP 4
347
348     PARAMETERS px[i+2] = C+AA+BB/2,
349                px[i+3] = C+AA+BB/2
350
351 NEXT i
352
353 !!! x-coordinates for 2nd and 3rd to final set of vertical panels:
354
355 FOR i=17 TO 111 STEP 4
356
357     PARAMETERS px[i+2] = BB+AA,
358                px[i+3] = BB+AA
359
360 NEXT i
361
362 !!! x coordinates for final set of vertical panels:
363
364 FOR i=113 TO 128 STEP 4
365
366     PARAMETERS px[i+2] = BB/2+AA+D,
367                px[i+3] = BB/2+AA+D
368
369

```

FIG. 13: Sample GDL code showing parametric expressions used to define classical proportions.

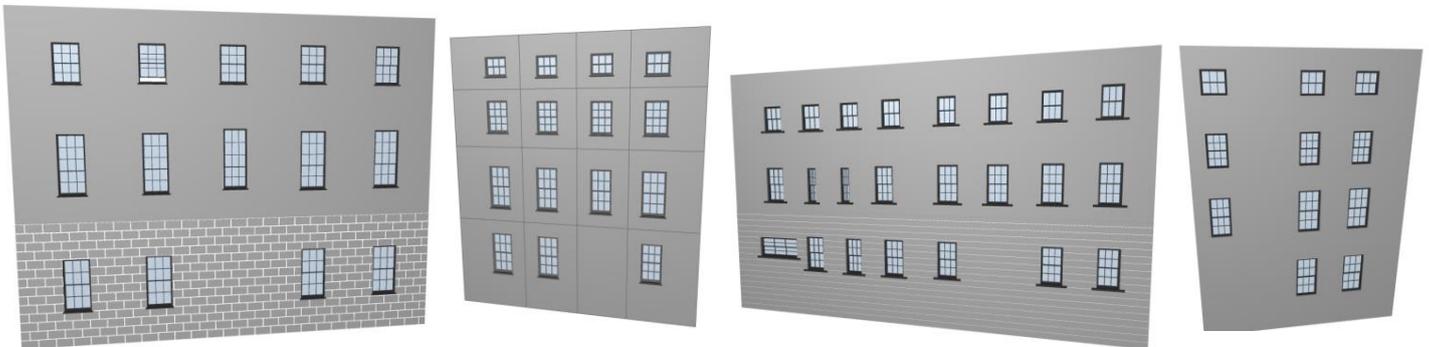


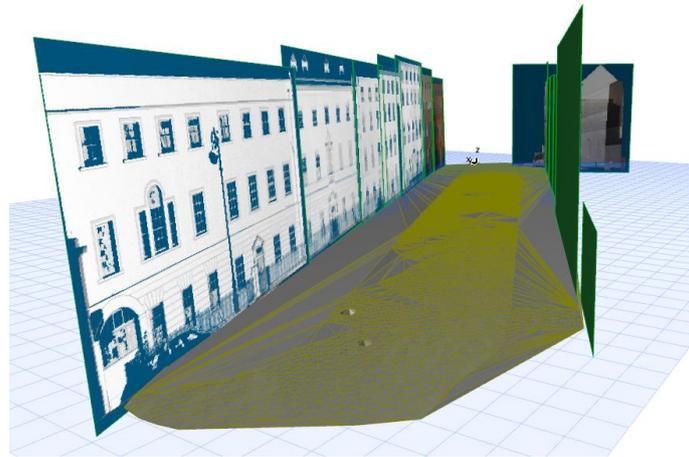
FIG. 14: Various façade models automatically generated by altering parameters of the façade template.

## 5. MAPPING TO SURVEY DATA

### 5.1 Integrating Survey Data

The developed parametric façade can be used for fast and efficient modelling of existing buildings with the ArchiCAD BIM software. Survey data obtained from laser scanning or photogrammetry can be used as source data for generating accurate façade models. Modelling directly to large point clouds obtained from laser or image data can be difficult as large point clouds can be very intensive on processing within BIM software which is not

designed for such large datasets. There can also be accuracy issues when modelling directly to point clouds in 3D space as it can be difficult to locate an object's exact position within a dense 3D point cloud. Instead of modelling directly to a complete point cloud, segmented point clouds, orthographic images in elevation and plan and 2D cuts through a point cloud are used to generate as-built BIM models. All survey data, including geo-referenced orthographic images can be imported in their correct 3D position in the ArchiCAD BIM software using its source coordinate reference system to avoid any alignment errors between datasets (FIG. 15).



*FIG. 15: Accurate integration of multiple survey datasets in 3D using a common coordinate reference system.*

## **5.2 Mapping Parametric Façade Template to Survey Data**

After all survey data is imported and correctly positioned in BIM software the newly developed parametric façade can be used for fast and efficient modelling of existing building façades. When the parametric façade is opened in BIM software it is first positioned in plan relative to survey data such as orthographic imagery created from the point cloud or mesh model in a plan view. When importing the parametric façade a number of initial parameters can be edited from a dialogue box. This includes parameters for the structure of a façade such as the wall thickness, number of storeys, number of horizontal tiles, door position and inclination angle of the façade if not perfectly vertical. The inclination angle can be accurately measured from orthographic imagery in the 3D window as all orthographic images are correctly positioned in 3D using coordinates of the image corners. By adjusting these initial parameters the required façade structure is automatically generated. A number of these parameters can also be graphically edited directly in the 2D and 3D windows by moving hotspots. This can be seen in FIG. 16 where parameters for the number of storeys and number of horizontal tiles are graphically edited to automatically generate the required façade structure.

The second stage of mapping the façade template to survey data is to apply the classical proportions to the façade model which provides an initial estimate for the position and size of façade elements. In order to apply these classical proportions two measurements are taken from the survey data and entered into the façade dialogue box as parameters. After these parameters are entered into the façade template the size and position of all elements on the façade model are immediately updated to reflect these classical proportions. When modelling classical historical buildings having these proportions already applied significantly reduces the amount of further editing to be carried out.

Once the structure of the façade has been generated the façade model can be overlaid with survey data and the initial estimates for objects provided by applying classical proportions can be assessed. The final stage then involves refining any elements on the façade that do not match the survey data. This involves graphically editing objects on the parametric façade model to accurately position elements relative to survey data. This is carried out in 2D or 3D while overlaying the generated model with survey data such as orthographic imagery (FIG. 17) and horizontal and vertical cut sections through the point cloud or 3D mesh model. In order to enable efficient

editing, editable hotspot points can move multiple objects at once. This simultaneous editing allows users to quickly alter the heights of all windows on a floor simultaneously, the width of all windows above each other in a column simultaneously and also the position of all windows in a column or floor simultaneously. The distance between floors and columns can also be modified graphically to move multiple objects at once. This ability to edit groups of objects simultaneously greatly speeds up the editing process. Along with this editing with groups of objects, individual editing of objects is also possible to achieve high levels of accuracy relative to the survey data. Individual window corners can be modified by simply selecting a hotspot at a window corner and moving it to the desired new position (FIG. 18).

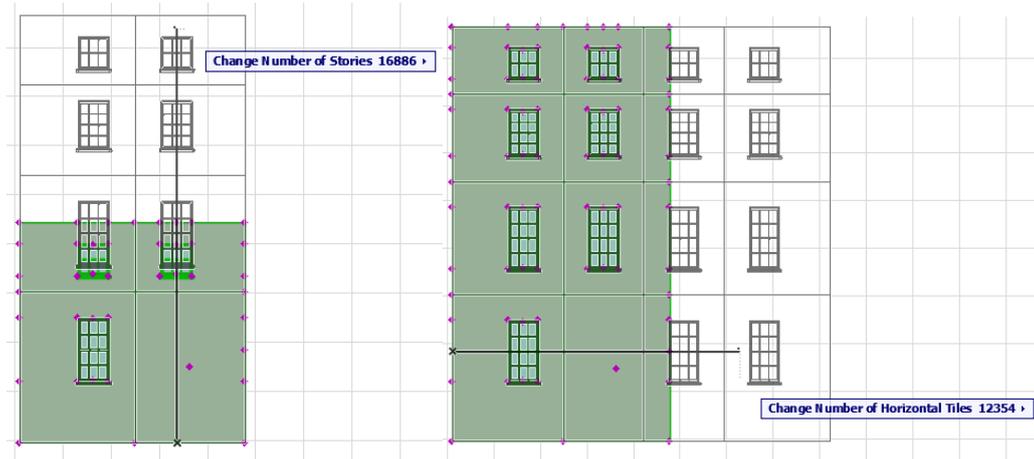


FIG. 16: Graphical parameter editing to procedurally generate structure of façade. Parameter for number of storeys edited (left) and number of horizontal tiles edited (right).



FIG. 17: Orthographic image overlaid with parametric façade template for graphical editing of façade elements.

As an element on a façade is modified all linked elements are also automatically updated as seen in FIG. 18 where graphically editing a window opening automatically updates the combined ashlar block wall detail which calculates the required subtractions or additions using Boolean operations.

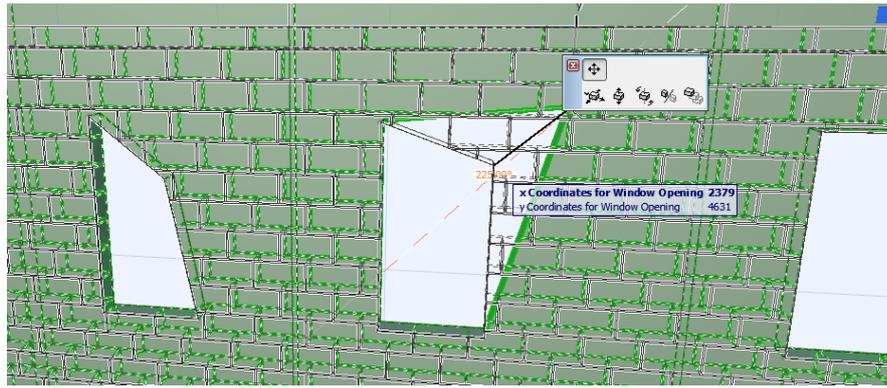


FIG. 18: Graphically editing window opening which automatically updates linked elements such as ashlar block wall detail.

Using this new parametric façade template it is possible to generate accurate façade models for existing buildings from survey data very quickly. Users first input the initial parameters to automatically generate the required façade structure. Next users input two parameters to apply classical proportions as an initial estimate for objects on the facade. Finally objects on the façade can be refined to accurately map the façade elements to survey data using simultaneous or individual graphical parameter editing.

### 5.3 Manual Mapping of Parametric Library Objects to Survey Data

Along with the methods for semi-automatically plotting building facades, manual plotting methods can also be used with existing individual HBIM library objects. The approach used with HBIM is to map the library objects in 2D onto segmented point clouds and orthographic images in elevation, plan and section. Before positioning a library object in the BIM software the default parameters of the object can be edited, changing the object's shape, size or other properties to correspond with the survey data. When library objects are brought into BIM software they are first positioned in plan using orthographic imagery or cut sections through the point cloud. The height of the object is specified by a parameter for its formation level. Objects are then more precisely positioned in front and side elevation from further orthographic imagery and cut sections. The next section shows an example of a point cloud and subsequent BIM model created using HBIM parametric library objects and the parametric façade described in this article.

## 6. CASE STUDY

A case study has been undertaken to show the application of the HBIM library objects and the parametric façade for modelling a heritage site with BIM software. The heritage site chosen is an 18<sup>th</sup> century Georgian Street located in Dublin, Ireland. The street known as Henrietta Street is one of the earliest Georgian streets in Dublin and is of great historical significance. The street was developed by Luke Gardiner and constructed between 1730 and 1820. The buildings on Henrietta Street show great classical style architecture and were originally seen as city palaces. During the 19<sup>th</sup> and 20<sup>th</sup> century's the street fell into disrepair and despite recent restoration work there are still a number of buildings that need urgent attention.

### 6.1 Data Collection

In order to show the efficiency of modelling with HBIM library objects and the parametric façade the entire street was recorded using laser scanning and image acquisition methods. Eight scans were carried out with 10mm resolution using a Trimble GS200 terrestrial laser scanner as seen in FIG. 19. Common targets were surveyed at a higher resolution (2mm) in common areas between scans for registration of scans. Five to eight common targets were surveyed between scans to ensure accurate registration results during processing.



FIG. 19: Trimble GS200 laser scan used to record data for generating as-built BIM model of street.

## 6.2 Data Pre-Processing

A number of pre-processing steps were carried out on the scan data before modelling with BIM software. These pre-processing steps included registration, segmentation/filtering, triangulation, texturing, orthographic image creation and the generation of cut sections. All processing of scan data was carried out using Trimble Realworks software. FIG. 20 shows the final point cloud recorded for Henrietta Street.

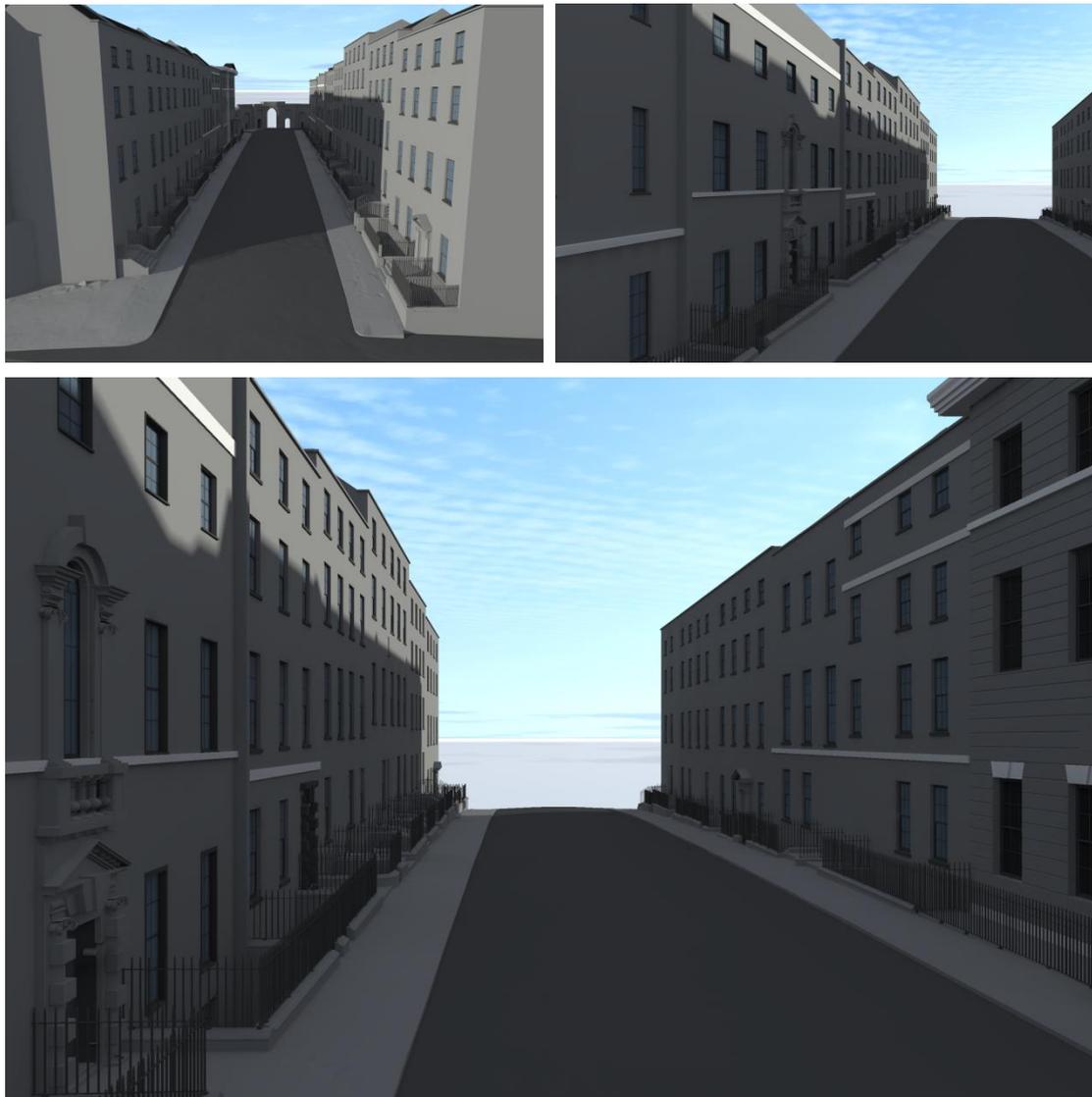


FIG. 20: Henrietta Street, Dublin (left) and point cloud of street coloured by intensity (right).

## 6.3 BIM Modelling with Parametric Façade and Library Objects

Next data from the survey including segmented point clouds, orthographic images and cut sections were imported into the ArchiCAD BIM software for modelling. The parametric façade and HBIM library objects were then used to generate as-built BIM models from the survey data. Manual modelling of each individual façade on this street would be a very time consuming task and would involve measuring, positioning and adjusting hundreds of objects to reconstruct the street. Instead each façade structure was automatically generated with the

parametric façade tool. Once each façade was automatically generated the position of façade elements were quickly refined using efficient simultaneous editing or individual editing if required. Additional HBIM parametric library objects were also added where required to complete the model. These HBIM parametric library objects were manually mapped to survey data and façade models. Future work will involve incorporating more HBIM parametric library objects to the parametric façade template to avoid the need for manual positioning of extra parametric objects. Instead these parametric library objects will be selected and automatically added to the parametric façade. FIG. 21 shows the BIM model generated with the parametric façade and HBIM library objects. After the model is created automatic documentation can be created in the form of plans, elevation and sections (FIG. 22).



*FIG. 21: BIM model created using HBIM library objects and parametric façade for Henrietta Street, Dublin.*



FIG. 22: Automated 2D documentation from BIM model.

## 7. EVALUATION AND TESTING

Preliminary testing has been carried out to evaluate the current prototype of the parametric façade plug-in for ArchiCAD BIM software. The aim of this evaluation is to test the usability and ease of use of the software plug-in and also to evaluate the contribution and potential benefits of the new plug-in. This section outlines the preliminary testing carried out to date and initial results from these tests. Future work will involve undertaking more comprehensive testing and evaluation of the newly developed plug-in for modelling existing buildings. The preliminary testing carried out will also assist with the design and implementation of future tests and evaluation.

### 7.1 End-User Scenario Testing

As part of an initial evaluation, end-user scenario testing has been carried out. End-user scenario testing involves creating test scenarios which replicate a typical end users usage of a software tool. This type of scenario testing evaluates the entire workflow of a program and can be used to find defects and return valuable feedback to validate and evaluate a program on key areas such as usefulness, usability and efficiency. Scenario-based testing is a conventional method used in software testing to identify problems and potential for improvement with a programme (Carroll 2000). Similar approaches have been adopted for software testing by Moore (2013) and Murphy (2012).

The specific aim of the end-user scenario testing implemented was to assess the usability and efficiency of the current parametric façade prototype for generating façade models of existing buildings and also to assess the quality of the façade models generated with the plug-in.

#### 7.1.1 Participants for End-User Scenario Test

The participants for the preliminary end-user scenario tests were chosen from a group of full-time third level construction and surveying students attending the Dublin Institute of Technology. As part of their studies all participants had completed modules and were experienced in 3D CAD, BIM, surveying and building maintenance and conservation. The group of participants were selected based on their experience in these areas and were a suitable representative of the end-user profile for the new plug-in. Future end-user scenario testing will include participants from both academia and industry.

#### 7.1.2 Testing Scenario

A test scenario was designed to simulate typical end-user usage of the parametric façade plug-in. The test scenario designed required participants to accurately model the façade of a real classical building, number three

Henrietta Street from survey data using the new parametric façade plug-in for ArchiCAD BIM software. The survey data included various products from a laser scan survey such as geo-referenced orthographic imagery in plan and elevation, segmented and complete point clouds and 2D cut sections through the point cloud. The test scenario first required participants to import and position the required survey data within the BIM software. Next the participants were asked to generate a BIM model for the building façade from survey data. In order to assess the usability and efficiency of the new parametric façade plug-in participants were required to carry out the test scenario twice. The first time participants were asked to create the BIM façade model using existing tools within the BIM software. This involves manual generation of the BIM façade geometry. The second time, participants were asked to create the BIM façade model again but this time using the new parametric façade plug-in. This allowed participants to compare the new developed techniques for semi-automatic façade modelling with existing techniques.

### 7.1.3 User-Evaluation Session

The user-evaluation session took place in a computer laboratory in the Dublin Institute of Technology. Before the scenario tests were carried out a presentation was made to the participants providing an overview of the research project to provide background and context. This presentation was then followed by a demo of the parametric façade plug-in. Participants then carried out both test scenarios which involved creating a BIM façade model using existing manual techniques and then again using the newly developed semi-automatic techniques. After completing both test scenarios participants were then asked to complete an online questionnaire to provide feedback and evaluate the plug-in that was tested.

The first part of the questionnaire was used to establish how much experience and with what 3D CAD and BIM software the participants had experience with. The next part of the questionnaire asked the participants to compare the efficiency and usability of the new semi-automatic process for generating façade models compared to existing manual methods. A five category scale was used to measure the level of agreement or disagreement with a statement or question to assess the efficiency and usability of the parametric façade plug-in. The next part of the questionnaire was used to get feedback on the methods of editing for the parametric façade and also the preferred survey data to use when generating an as-built façade model. The methods of editing include graphical editing of parameters directly in 2D or 3D and manual input of parameters in a dialogue box. The final part of the questionnaire was used to get feedback on the quality of the façade models that could be generated with the new plug-in. A five category scale was used again to determine the level of satisfaction or dissatisfaction with the quality and level of detail of the generated façade models.

### 7.1.4 Results

In the first part of the questionnaire participants were asked which software they had used in the past for 3D modelling. The results showed that participants had experience in a range of 3D CAD and BIM software including Sketchup (40%), ArchiCAD (20%), AutoCAD (20%) and 123D Catch (20%)(FIG. 23). Next participants were asked to rate the usability of the parametric façade plug-in compared to existing modelling methods with 3D CAD and BIM software. The results indicate that 33% of participants found the parametric façade plug-in equally as difficult or easy to use as existing 3D CAD/BIM methods, 33% found it easier than existing 3D CAD/BIM methods and 33% found it much easier than existing 3D CAD/BIM methods (FIG. 24).

FIG. 25 shows results relating to the efficiency of the parametric façade plug-in. For this users were asked to rate the efficiency of modelling with the parametric façade plug-in compared to existing 3D CAD/BIM methods. The results for this indicate that 44% of participants found the parametric façade plug-in to be more efficient than existing 3D CAD/BIM methods and 56% found it much more efficient than existing 3D CAD/BIM methods.

FIG. 26 shows results relating to the quality of the models generated with the parametric façade plug-in. Participants were asked to rank on a scale of 1 (very poor) to 5 (very good) the quality of the 3D façade model generated with the parametric façade plug-in. 40% of participants rated the quality of the generated model a factor of four and 60% rated the quality a factor of five which indicates that participants found the quality of the façade model and objects in the façade to be very good. More feedback from the questionnaire showed that 80% of participants found graphical editing to be more efficient and more usable than editing parameters from a dialogue box. When participants were asked to specify their preferred survey data for modelling existing buildings, 50% selected orthographic imagery, 10% selected 3D vectors, 10% selected data sheets and 30% selected a combination of sources.

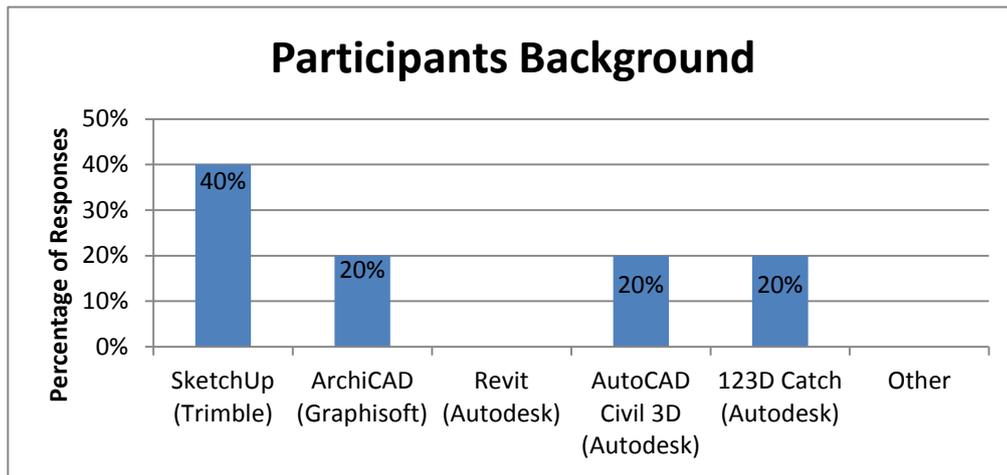


FIG. 23: Results of end-user scenario testing relating to the 3D CAD/BIM software that participants were experienced in.

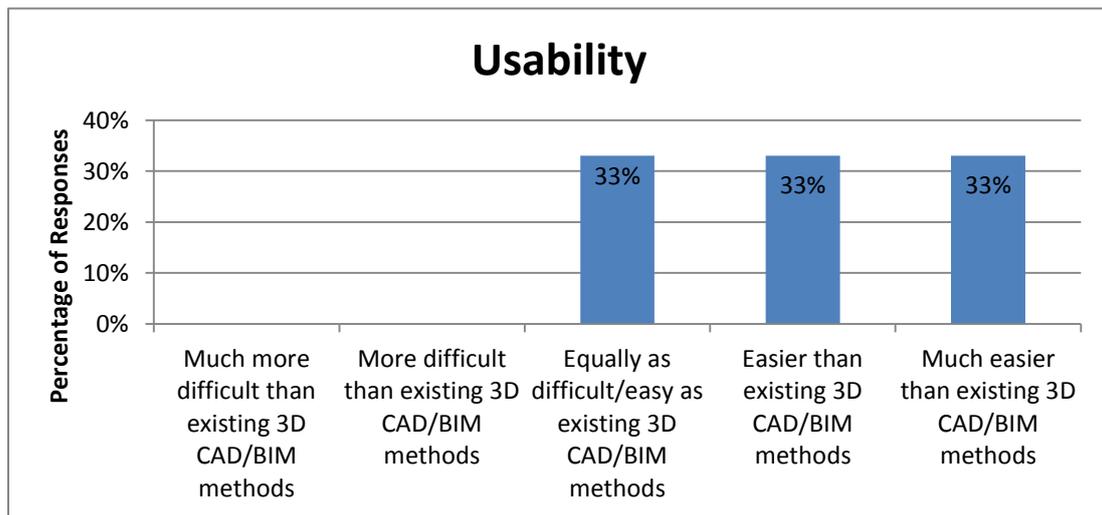


FIG. 24: Results of end-user scenario testing relating to the usability of the parametric façade plug-in.

### 7.1.5 Review of Results

These results from preliminary end-user scenario tests indicate that users found the semi-automatic approach with the parametric façade plug-in to be more efficient than existing methods for creating building façade models. The majority of participants also found the parametric façade plug-in to be easier to use than existing methods. A user does not need to be very experienced in 3D modelling to create a façade model with the parametric façade. Instead of building the model from scratch users simply need to adjust parameters to match survey data. This testing also indicates that the level of detail and quality of the elements in the current prototype of the façade model are very good. Additional feedback from this end-user scenario testing has shown that graphical methods for editing parameters are preferred over traditional methods for editing parameters from a dialogue box. As a result of this feedback future work will involve making more parameters editable with graphical methods. Further informal feedback was also obtained during discussions with participants throughout the evaluation session. One of the main limitations that participants mentioned about the current prototype for the parametric façade was that more parametric library object should be available to automatically link to the

current parametric façade template. Future work to the parametric façade plug-in will involve incorporating more HBIM library objects to make the parametric façade more useful for end-users. The feedback obtained from the evaluators throughout the evaluation session provided evidence of the potential benefits, shortcomings and future applications of the implemented parametric façade tool.

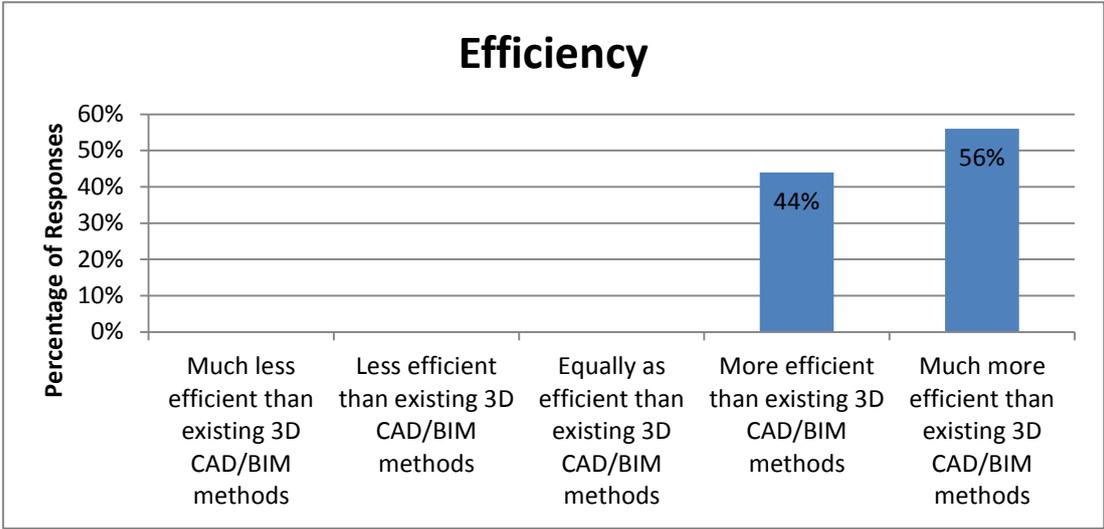


FIG. 25: Results of end-user scenario testing relating to the efficiency of the parametric façade plug-in.

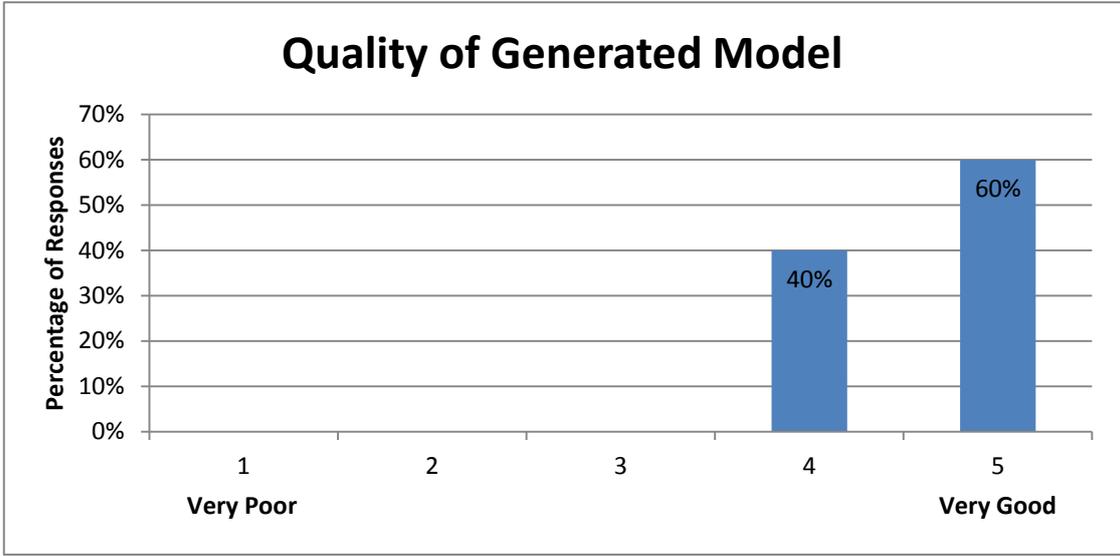


FIG. 26: Results of end-user scenario testing relating to the quality of the model generated with the parametric façade plug-in.

### 7.2 Accuracy of Façade Plot against Survey Data

Along with the end-user scenario testing a preliminary evaluation has also been carried out to test the accuracy of BIM façade models generated with the new parametric façade plug-in. The accuracy of a façade model was measured by comparing a sample of measurements from a BIM façade model generated with the parametric

façade plug-in to related points from the original point cloud that was captured to record the existing building. A total of thirty coordinates were measured on a BIM façade model, which was generated with the new plug-in from orthographic imagery and cut sections through the point cloud. This was a random sample of defined points that were well distributed throughout the façade model. The x, y and z coordinates were recorded for each of the points selected on the BIM façade model. The same related points were then measured on the original point cloud using the Trimble Realworks point cloud processing software. The x, y and z coordinates were recorded for the same thirty defined points on the point cloud. As both the point cloud and generated BIM façade model were referenced to the same coordinate reference system, the resulting coordinates could be compared directly.

The results are outlined in TABLE 3. The mean difference between the coordinates recorded on the BIM façade model and the point cloud was 0.010m in x, 0.012m in y and 0.005m in z. The standard deviation between the coordinates recorded on the BIM façade model and the point cloud was 0.013m in x, 0.017m in y and 0.006m in z.

*TABLE 3: Comparison between coordinates recorded on a BIM façade model generated with new plug-in and original point cloud.*

	<b>Difference in x-coordinates</b>	<b>Difference in y-coordinates</b>	<b>Difference in z-coordinates</b>
<b>Mean Error:</b>	0.010m	0.012m	0.005m
<b>RMSE:</b>	0.010m	0.012m	0.005m
<b>Standard Deviation:</b>	0.013m	0.017m	0.006m

These results shown in TABLE 3 are within the recommended tolerances from the English Heritage (Bryan et al 2009) for recording heritage objects of that size. As shown in TABLE 4, a tolerance of +/-15mm is recommended by the English Heritage for recording small structures approximately 20 metres by 30 metres in size. The resulting mean and root mean squared errors from the accuracy tests were all within the recommended +/-15mm tolerance. The standard deviation for x and z coordinates were also within the recommended tolerance however, the standard deviation of 0.017m for the y-coordinates was slightly over the recommended tolerance.

One possible cause for errors can be due to the subjective nature of identifying points from survey data such as point clouds or orthographic images. Precise corners can often be difficult to define due to decay of materials at edges or corners of an object. Scanning objects at higher point densities can help reduce these errors but will lengthen the survey time.

*TABLE 4: Density of point cloud and measurement precision (Bryan et al 2009)*

<b>Scale</b>	<b>Point Density</b>	<b>Precision</b>	<b>Typical Use</b>
1:10	2mm	+/-2mm	Small objects 5mx5m
1:20	4mm	+/-4mm	Large objects 10mx10m
1:50	15mm	+/-15mm	Small structures 20mx30m
1:100	25mm	+/-25mm	Large structures 40mx60m

## 8. CONCLUSION

This article has presented a new approach for semi-automatic modelling of building facades from laser or image data using a parametric façade template which has been developed as a plug-in for the ArchiCAD BIM software. This methodology incorporates concepts from shape grammars for the design and implementation of the parametric façade object. The parametric façade template facilitates fast and efficient modelling of endless configurations of building facades. A building façade structure is first procedurally generated by altering parameters for the number of storeys, number of horizontal tiles and the door position on the ground floor. When automatically generating the façade, the initial position and size of elements are estimated using classical architectural proportions. After the façade is automatically generated users can then interactively edit the position, size and other parameters of façade elements to accurately map objects to survey data. The parametric façade template has been implemented with the Geometric Description Language for ArchiCAD BIM software. This enables the tools developed to utilise the full benefits of BIM software which includes automated construction or conservation documents, semantic object oriented objects based on IFC semantic classes, automatic lists of objects and material and the ability to add and link additional information to the model.

Initial tests have shown that the parametric façade is more efficient than existing methods for creating façade models from survey data. The façade template also provides an easier solution for generating façade models when compared to existing manual methods. Non-specialist users who may not have a lot of experience in 3D modelling can easily generate and modify the façade template by altering parameters graphically or from a dialogue box.

## 9. FUTURE WORK

Future work will involve extending the parametric façade with more parametric library objects such as additional window types, various door types and door case detail from classical orders including columns, architrave detail, entablatures and pediments. Many of these objects have already been coded as part of the HBIM parametric library which can be incorporated into the parametric façade. Having these detailed objects incorporated into the parametric façade would allow for automatic generation of a typical façade in a particular classical style such as Doric, Corinthian, Tuscan or Ionic. Users could then make modifications to the generated façade to match to survey data when creating as-built BIM models.

Future work will also focus on a more procedural approach for generating entire building models and not just façade models. This will maintain the powerful parametric control over objects to allow for fast and efficient reverse engineering of existing buildings from survey data.

Further testing will also be carried out which will involve more comprehensive end-user scenario testing with participants from academia and industry. Further accuracy tests will also be carried out on a variety of building types with independent measurements taken on buildings using total station equipment. Future work will also investigate designing similar plug-ins for other BIM software such as Revit from Autodesk.

## 10. REFERENCES

- Boeykens S. (2011). Using 3D design software, BIM and game engines for architectural historical reconstruction, paper presented to CAAD futures 2011, Liège, Belgium, 6-8 July 2011.
- Boeykens S., Himpe C. and Martens B. (2012). A case study of using BIM in historical reconstruction - the vinohrady synagogue in prague, paper presented to The 30th international conference on education and research in computing aided architectural design in Europe, Prague, Czech Republic, 12-14 September 2012.
- Bryan P., Blake B., Bedford J., Barber D., Mills J. & David A. (2009). Metric survey specifications for cultural heritage, English Heritage.
- Carroll J.M. (2000). Making use: scenario based design of human-computer interactions, The MIT Press, Cambridge.
- Chevrier C., Charbonneau N., Grussenmeyer P. and Perrin J.P. (2010). Parametric documenting of built heritage: 3d virtual reconstruction of architectural details, *international journal of architectural computing*, vol. 08, no. 02, pp. 131-45.

- Fai S., Graham K., Duckworth T., Wood N. and Attar R. (2011). Building Information Modelling and heritage documentation, paper presented to XXIII CIPA international symposium, Prague, Czech Republic, 12th-16th September.
- Graphisoft (2011). *GDL Reference Guide*.
- Hichri N., Stefani C., Luca L.D. and Veron P. (2013). Review of the as-built BIM approaches, paper presented to 5th international workshop, 3D-ARCH-2013. 3D virtual reconstruction and visualisation of complex architectures, Trento, Italy.
- Hohmann B., Krispel U., Havemann S. and Fellner D. (2009). CityFit - high-quality urban reconstructions by fitting shape grammars to images and derived textured point clouds, paper presented to 3D-ARCH 2009 - 3d virtual reconstruction and visualization of complex architectures, Trento, Italy, 25-28 February 2009.
- Lipp M., Wonka P. and Wimmer M. (2008). Interactive visual editing of grammars for procedural architecture, *ACM transactions on graphics*, vol. 27, no. 3, p. 1.
- Moore N. (2013). 'A model based system for contextual on-site construction planning in augmented reality', University of Wolverhampton.
- Müller P., Wonka P., Haegler S., Ulmer A. and Gool, L.V. (2006). Procedural modeling of buildings, *ACM transactions on graphics*, vol. 25, pp. 614-23.
- Müller P., Zeng G., Wonka P. and Gool L.V. (2007). Image-based procedural modeling of facades', *ACM trans. graph.*, vol. 26, no. ACM, p. 85
- Murphy M. (2012). 'Historic Building Information Modelling (HBIM), for recording and documenting classical architecture in Dublin 1700 to 1830', Trinity College Dublin.
- Murphy M., Govern E.M. and Pavia, S. (2013). Historic Building Information Modelling – adding intelligence to laser and image based surveys of european classical architecture, *ISPRS journal of photogrammetry and remote sensing*, vol. ISSN 0924-2716.
- Pauwels P., Verstraeten R., Meyer R.D. and Campenhout J.V. (2008). *Architectural Information Modelling for virtual heritage application*, digital heritage -- proceedings of the 14th international conference on virtual systems and multimedia, Archaeolingua.
- Stiny G. (1977). Ice-ray: a note on the generation of Chinese lattice designs, *environment and planning B*, vol. 4(1), pp. 88-98.
- Stiny G. and Gips J. (1972). Shape grammars and the generative specification of painting and sculpture, *the best computer papers of 1971*, pp. 125-35.
- Tang P., Huber D., Akinci B., Lipmand R. and Lytle A. (2010). Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques', *automation in construction*, vol. 19, no. 7, pp. 829-43.
- Thaller W., Krispel U., Havemann S., Redi I., Redi A. and Fellner D.W. (2011). Developing parametric building models - the gandis use case, paper presented to 4th international workshop 3D-ARCH 2011, "3d virtual reconstruction and visualisation of complex architectures", Trento, Italy, 2-4 March.