

# Parametric Documenting of Built Heritage: 3D Virtual Reconstruction of Architectural Details

Christine Chevrier, Nathalie Charbonneau,  
Pierre Grussenmeyer and Jean-Pierre Perrin



# Parametric Documenting of Built Heritage: 3D Virtual Reconstruction of Architectural Details

Christine Chevrier, Nathalie Charbonneau, Pierre Grussenmeyer and Jean-Pierre Perrin

## **ABSTRACT**

This paper examines 3D modelling of architectural elements with the help of parametric components. Such components may be useful within the framework of projects dealing with virtual 3D reconstruction of heritage monuments. Architectural components of the built heritage often have complex geometry. We studied the various geometrical shapes of a given architectural element, representative of a specific period and place. This study allowed us to identify the parameters and to implement parametric objects (in Maya Environment [1]). We also developed a Graphical User Interface (GUI) to answer the user's needs while generating the 3D model representing the architectural element. Within this GUI, the user is able to make adjustments with the help of laser point clouds from laserscanning, 2D plans or photographs. We exemplify our method with a case study dealing with openings and lintels. The corpus under study consists of elements of the built heritage of Montreal (Canada) and Nancy (France).

## I. Introduction

Data acquisition techniques today produce accurate 3D data sets. Thanks to these technological advances, 3D models are now used in various fields such as scientific and architectural studies. For instance, 3D data sets are quite useful in research projects dealing with virtual 3D reconstruction of heritage monuments. It is possible to generate images illustrating built environments, to elaborate 3D models enabling virtual walkthrough (so as to better understand the configuration of a monument), and so forth.

However, while the data is readily available, the modelling process is still time consuming. On the one hand, buildings are almost always characterised by a complex geometry. On the other hand, modelling architectural details is no easy matter. Indeed, when tackling the problem of describing their geometry, the elaboration of a distinct 3D model for each one of the existing scenarios can be a tedious task.

We argue that an approach based on reusable models would optimize the modelling process. We propose to automate the step of the modelling process dealing with architectural details.

Within the framework of this research project, we focus on the elements adorning the openings in heritage buildings. We work mainly on lintels, which are both structural and ornamental elements making up part of the opening. Our main aim is to demonstrate how it would be possible to systemize the generation of 3D models illustrating various interrelated scenarios.

Our aim is to develop a digital tool for virtual 3D reconstruction of heritage monuments. It is important to emphasize that the current research project is a work in progress. We work on lintels with the object of assessing the potential of our approach and the efficiency of our methods. The relevance of such an experiment lies in our ability to develop the system beyond the case study dealing with lintels. We will implement the digital tool stone by stone so to speak. Our experiments on lintels will lead us to work on other architectural elements (different types of openings and other elements of the building front) so as to extend the scope of the digital environment.

Rather than modelling the building from scratch, we propose a method based on parametric objects whereby elements are first created and then adapted to the measured data. This method enables the user to readily model the architectural element and, afterwards, make the required adjustments with accuracy. These adjustments may be based on data sources such as plans, photographs and/or point clouds (generated by means of photogrammetric techniques or 3D laser scanning).

After discussing related works in part 2, part 3 explains the principles and advantages of our method. In part 4, we explain how we handle the description of the openings so as to generate 3D models illustrating various interrelated scenarios. Part 5 presents the case study dealing with lintels.

Part 6 explains the implementation of the parametric objects and how one can adjust a 3D model so as to make it fit with the measured data. Some examples are exposed in part 7 and, finally, we conclude and describe future work in part 8.

## 2. Related works

With regard to built heritage, various methods exist to acquire 3D data of existing buildings. The purpose of such techniques is to generate point clouds. Using these, it is then possible to automatically generate new numeric data such as dimensions or new geometric data such as lines or 3D primitives. Making use of photogrammetric techniques [2–4], it is possible to automatically retrieve the dimensions of an architectural element. The aforementioned methods developed by the researchers (Debevec et al., El-Hakim et al.) enable the user to gather the information required for modelling the main parts of a building. Nevertheless, for detailed and complex objects (such as volutes or vaults), a great number of pictures and/or corresponding points are required, thereby increasing the time devoted to the modelling process.

Mueller et al. [5] propose a method to readily model façades from upright pictures. Horizontal and vertical straight lines are computed. This automated process leads to the creation of simple shape (box) elements. Afterwards, the user provides the numerical values that will be assigned to the depth of the various elements. It is worth mentioning that, even though this method is suitable for simple modern building fronts, it may not be adaptable to more complex architectural elements such as neoclassical façades or architectural details.

A third method, involving lasergrammetry techniques [6,7] produces point clouds with very high density and precision. From these 3D data sets, it is possible to automatically generate geometrical primitives. However, only simple forms can be automatically created (such as sphere, cube, plan, cylinder, etc.). Boulassal et al. [8] extract, from laser point clouds, the lines defining the configuration of building fronts and can therefore establish the dimensions of openings and walls. Tests were made on rather simple façades in comparison with neoclassical buildings; that is to say the composition of the building fronts did not include architectural elements such as lintels, cornices, pediments and so forth.

When dealing with built heritage, we believe that it would be worthwhile considering other approaches liable to complement the methods based on the generation of numerical values and simple geometrical forms. In order to be able to take into account architectural elements and details characterised by a complex geometry, we propose to resort to parametric objects. Such objects are now available in libraries for specific fields (steel structure, mechanics, etc.) but, at the present time, it seems that there exist no libraries of parametric architectural components related to the built heritage. We believe that this new approach could enable

us to better handle the geometric complexity of architectural elements, while profiting from the advantages inherent in photogrammetric techniques.

It bears mentioning that the usefulness of parametric objects goes beyond the virtual reconstruction of the built heritage. It could prove an interesting tool for the architectural designer [9] or can be used to explore new architectural shapes within gaming or virtual worlds.

### 3. Principles of the method

We propose to elaborate a library of 3D parametric objects describing architectural elements from the built heritage. Previous research work led our team to explore innovative ways of generating 3D models of architectural elements such as columns, copulas, etc. The description of the geometry of these elements is based on parameters [10]. We now aim to further our investigation by producing accurate, consistent and re-usable models. In order to do so, we are currently implementing a digital tool. The purpose of this tool is to reduce the time involved in the modelling process during virtual 3D reconstruction projects.

The elements making up such a library, and also the range of scenarios included within each object, arise from our analysis of an existing built environment. We selected various configurations representative of a given built environment and classified them by type. This procedure was based solely on our observations and did not refer to architectural rules excepted from old treaties dealing with ancient and classical styles.

Modelling ancient monuments from these kinds of rules undoubtedly appears to be a promising field of research [11,12]. However, such rules only apply to a rather small range of elements (such as mouldings and columns typical of the neoclassical period). When tackling the task of modelling heritage buildings dating from more recent periods, these rules are totally useless; architectural forms follow from previous ones, in order to adapt to new contexts and/or new criteria as regard to aesthetic. This is the reason why we propose to organise the various occurrences of a given architectural element as a typology. We then analyse the typology in order to identify principles, rules of proportions and parameters that enable us to adequately develop parametric objects liable to generate 3D models illustrating every scenario within the sampling.

To elaborate a 3D model representing an architectural element, the user has to select from alternate options and provide estimated numeric values that will be assigned to the parameters. A 3D model can then be generated to display the virtual shape of one among various alternate scenarios. After making sure the generated model is in accordance with the scenario under study, the user can resort to point clouds to adjust the parameters' value with accuracy.

It is worth mentioning that models generated using this method are generic; they may not represent properly every aspect of the real object.

Since a window may not be a perfect rectangle, surfaces have to be “eroded”, so to speak. In those cases, deformers and sculpt geometry tools can be used to adjust the shape (transform a rectangle into a trapeze and so forth). It is then possible to adapt perfect mathematical models to imperfect shapes. Furthermore, sculpted elements modelled with meshes can be inserted in the scene description file library.

## 4. Openings

### 4.1. Geometrical study

As we said earlier, our aim is to tackle the problem of modelling architectural details characterised by a high level of geometric complexity (in comparison with modern architecture). With this in mind, we are currently working on a sampling of architectural details dating from the nineteenth and the beginning of twentieth century. Within the framework of a joint research project (involving researchers from both France and Canada), we undertook a case study dealing with openings in residential buildings in Nancy (France) and Montreal (Canada). Nancy can be twinned with Montreal because the types of openings are consistent with one another; many configurations of openings can be observed in one town as well as in the other. Thus, the digital environment we are developing is liable to be useful in both contexts.

Our approach stems from the fact that openings are compound objects. On the one hand, several architectural elements must be generated and combined to define the opening’s outline. There are upper elements (such as the lintel), lateral elements (such as abutment, attached pier, reveal, etc.) and lower elements (such as sill, breast-wall, etc.). On the other hand, there are elements dividing the opening (such as mullion, transom, etc.). For each one of the elements, different configurations coexist in both Nancy and Montreal (see Figure 1). Some of these elements (such as mullion or breast-wall) are optional whereas other elements are mutually exclusive. The constituent parts are interchangeable, making up various combinations.

▼ Figure 1. Windows from the built heritage of Nancy (on the left) and of Montreal (on the right).



## 4.2. Parameters

We are aiming to describe the object geometry with a minimum of parameters. For the sake of efficiency, it is important that we propose a coherent and intelligible organisation. Here are some of the parameters we have identified so far (please refer to Figure 2):

- elements which are in juxtaposition with the upper part of the opening (in dark yellow),
- elements which are in juxtaposition with the lower part of the opening (in light yellow),
- elements which are in juxtaposition with the right-hand side of the opening (in dark blue),
- elements which are in juxtaposition with the left-hand side of the opening (in light blue),
- upper shape of the opening, to be transmitted to the lower shape of the lintel (in dark red),
- lower shape of the opening (in light red),
- inner profile (in light green),
- outer profile (in dark green).

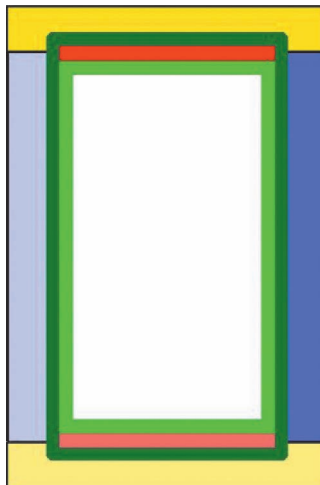
Since a great many configurations of windows exist, and as it is quite complicated to cope with a huge sampling, we decided to focus on one of the parts making up the opening. In the next section, we describe the analysis we made of the lintel and its constituent elements. By cutting down on the sampling, we were able to undertake a more thorough study.

## 5. Lintels

### 5.1. Geometrical study and typological chart

As we saw earlier, several architectural elements have to be generated and combined to create an opening. The same logic applies to the lintel.

► Figure 2. Parameters describing the opening.



The elements that go to make up the lintel are the lower elements (such as corbel pieces), the lateral elements (such as abutment), the upper elements (such as cornice) and the inserted elements (such as keystone). Again, the constituent elements are interchangeable, making up various combinations.

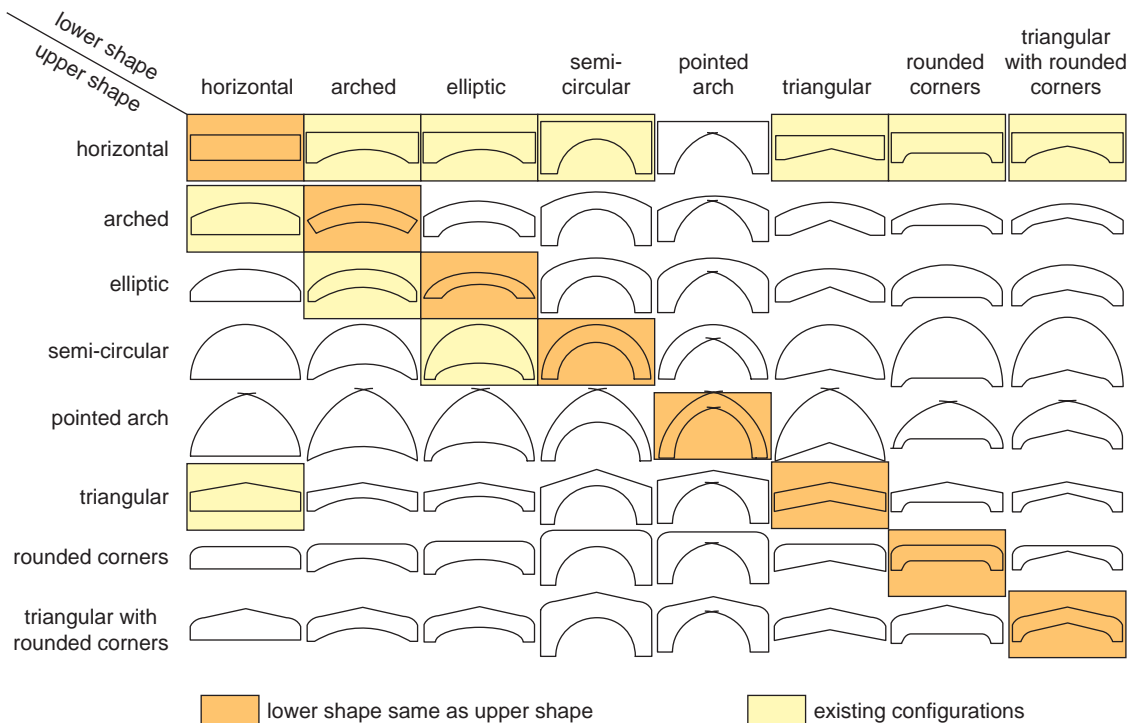
These constituent elements are combined with a central piece. This element, in its simplest form, is a parallelepiped rectangle. In this case, we consider that the upper shape and the lower shape are both rectilinear. Obviously, the shapes vary from specimen to specimen; they can be semi-circular, oblong, triangular, etc. In many instances, the shape of the upper part is the same as the lower part. Nevertheless, in other cases, one varies from the other one.

## 5.2. Parameters

By analysing our sample of lintels, we have identified the main following parameters:

- Width, height and depth (Figure 4.a),
- Shape of the upper part: horizontal, rounded, semi-circular, pointed arch, triangular, elliptic, rounded corners (see Figure 3), or other shape defined by a profile with the help of the graphical user interface,
- Shape of the lower part (which may be the same as the upper shape or one of the aforementioned cases),

▼ Figure 3. Lintels combining different lower shapes and upper shapes.



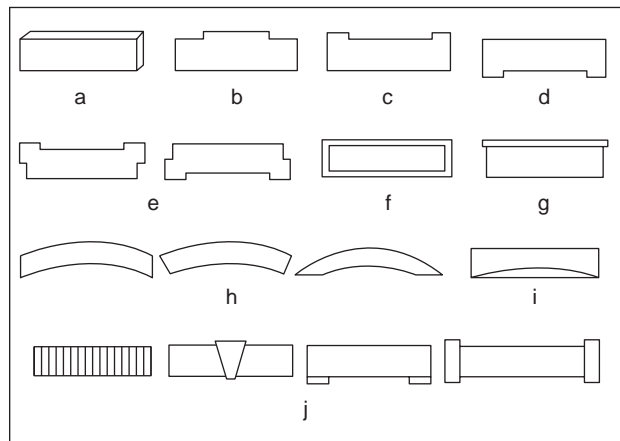


- Upper central offset: width and height (Figure 4.b),
- Upper lateral offset: width and height (Figure 4.c),
- Lower lateral offset: width and height (Figure 4.d),
- Lower and upper elbows (Figure 4.e),
- Profile surrounding the front face of the lintel (Figure 4.f),
- Upper cornice (Figure 4.g),
- Endings of the sides: vertical, horizontal, or perpendicular to the lower shape (Figure 4.h),
- Bevel with the following parameters: type, broach, depth, shape (Figure 4.i),
- Sub-elements: bricks, keys, corbels, abutment (Figure 4.j).

## 6. Implementation

Software development is performed in C++ and Mel (Maya embedded Language) [1] to create plug-ins (see part 6.1) and a graphical user interface for Maya (see part 6.2).

► Figure 4. Parameters describing the lintel.



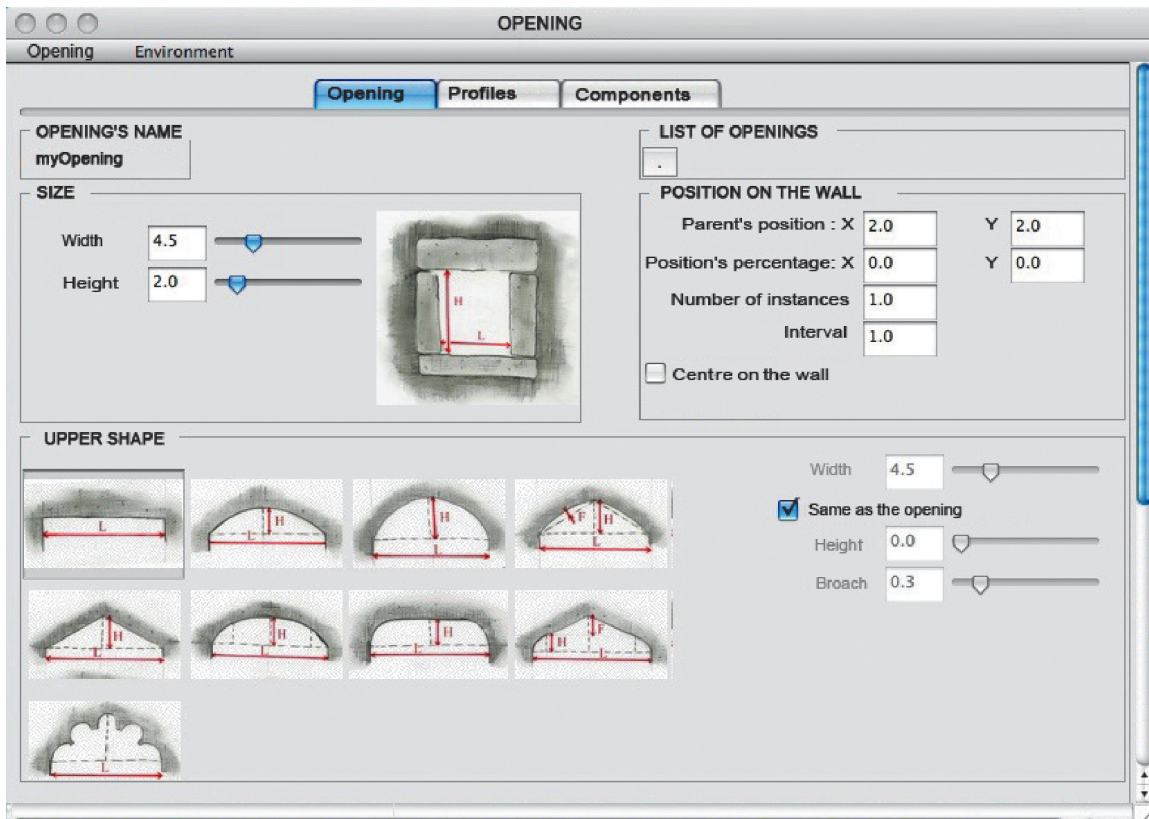
### 6.1. C++ classes

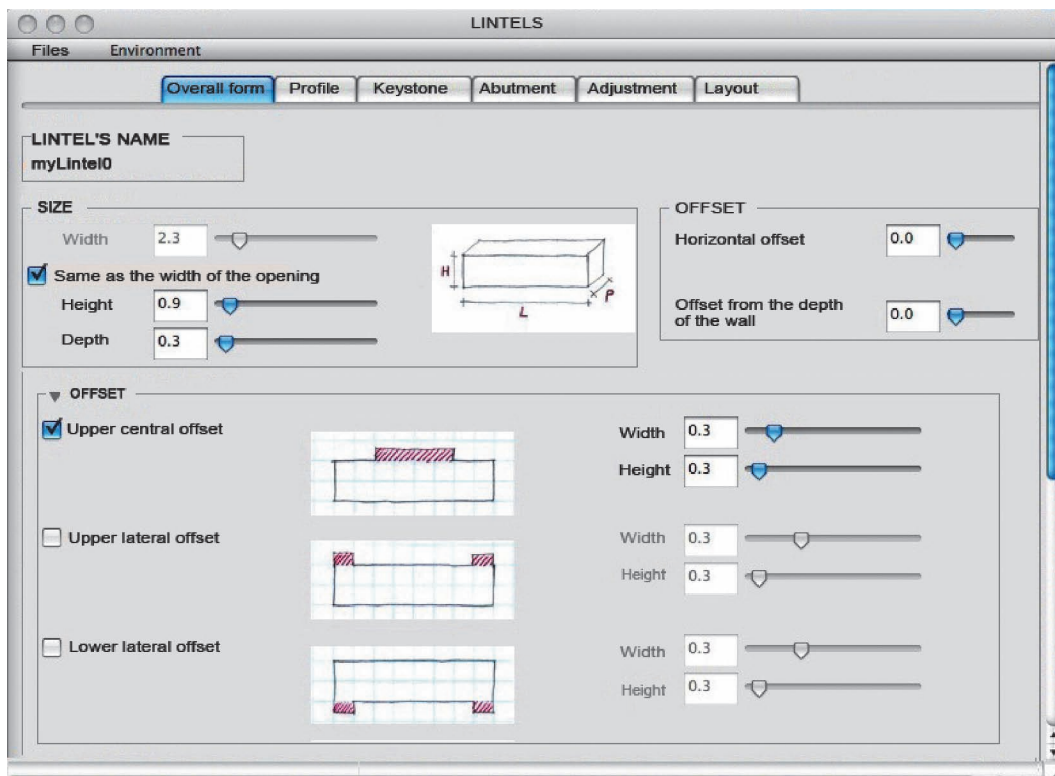
The digital environment is implemented making use of Maya's geometric modelling kernel. A Maya object is described via a set of attributes displayed in a window (the attribute editor). When an attribute is modified, the object is regenerated with the new parameters' values. To each one of the architectural elements (for instance the lintel) corresponds a Maya object; to each parameter (for instance the height) corresponds a Maya attribute. Objects are hierarchically organized, that is to say each element is composed of a set of sub-elements. As we said earlier, to elaborate a 3D model representing an architectural element, the user has to select from alternate options and provide estimated numeric values that will be assigned

to the parameters. After generating the 3D model of an architectural element, the user may decide to save, in a text file, the configuration of the scenario under study, for further use. In this file, architectural components are described with their own set of values. When this file is re-loaded, a plug-in reads the data and creates the corresponding Maya objects and the hierarchy. The values retrieved from the file are assigned to the parameters and, consequently, the 3D model is generated.

Apart from the parameters' value, these text files contain the hierarchical description of the scene. However, neither ascii file format nor Maya Embedded Language are useful in this context. Since both file formats are mostly made up of command lines, it is impossible to describe straightforwardly the creation of nodes or the hierarchical scene. That is the reason why we choose another description language, the Open Inventor file format (i.e. VRML 1), in order to describe the hierarchy and the parameters of architectural elements. This way, the structure of the components' hierarchy is adequately described and the value assigned to each parameter is clearly indicated. It is then possible to save text files containing this information, hence providing a library of scenarios. It bears mentioning that such a data base does not occupy much disk space (in comparison with 3D models).

▼ Figure 5. Section of the GUI used for defining the overall form of openings (profiles and components are defined in other tabs).





▲ Figure 6. Section of the GUI used for defining lintels (profile, keystone, abutments, adjustment and layout are defined in other tabs).

To ensure that the constituent parts of an opening are combined coherently, it is essential to formalize the connections between those components. In order to do so, we developed a C++ class named "Opening". This class positions each of its sub-elements (lower, upper, right and left components) according to the overall dimensions of the opening. Indeed, components' dimensions are linked together. When the dimensions of the opening vary, these modifications are transmitted to the sub-elements; if the width varies, the width of the upper and lower elements is automatically updated. If the height varies, the lateral sub-elements are modified. The same holds good for the lintel; the C++ class "Lintel" positions each of its sub-elements (corbel pieces, key, etc.) according to the overall dimensions of the lintel. Forms' specifications are also linked together. The upper shape of the opening can be automatically assigned to the lower shape of the lintel; when modelling an opening, the user chooses whether these shapes should be connected or not.

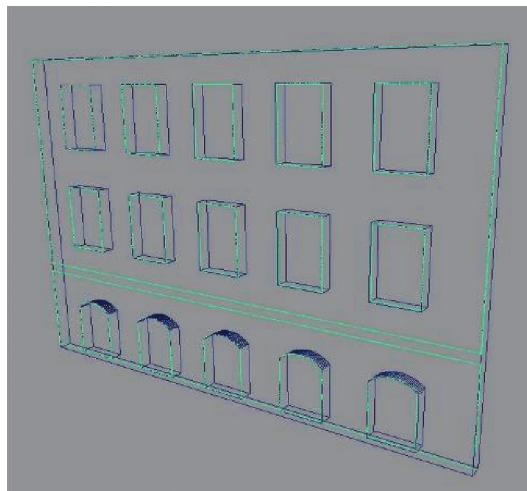
Since modifying the text file containing the parameters describing a given scenario could be rather tedious for the user, we developed a graphical user interface. This way, the user can interact with the system by means of windows containing radio buttons, sliders and text boxes. Two of those windows are presented below (Figure 5 and 6); the first one deals with an opening, the second one with the elements making up the lintel.

We put a lot of care into designing the graphical user interface. The problem which was uppermost in our minds was to find ways of enabling the user to easily navigate within the digital environment. We wanted him or her to be able to readily locate the controls in order to efficiently elaborate the 3D model and, afterwards, make the required modifications. For instance, when modelling a lintel, the user provides in turn the overall dimensions, specifications regarding upper and lower shapes, keystone, corbel pieces, etc. He or she can access each one of these elements by selecting a tab.

While configuring these tabs, our aim was to provide a user-friendly environment. We assumed that architectural vocabulary may not be well-known to everybody. With that in mind, we implemented radio-buttons enabling the user to make selections among sets of images illustrating the alternate options. Furthermore, we deemed necessary to provide a flexible way to input numerical data. At the beginning of the process, the user is required to provide estimated values which will be modified subsequently in order to make adjustments. He or she can provide numerical data by means of a text box and, later on, modify it by means of a slider. After each modification, the 3D model is regenerated in real time. So the user is able to verify whether the modification produces the desired effect on the shape.

## 6.2. Adjustment to measured data

When modelling an architectural element, the last step the user has to perform is to adjust the parameters so as to adapt the shape to the scenario under study. To assist the user while adjusting the size and proportions of the various components, we offer the possibility of using different types of data within the graphical user interface: 2D (such as pictures, plans or photographs) and 3D (such as point clouds).



► Figure 7. Parametric model of the wall and its openings automatically computed from laser terrestrial point clouds.

► Figure 8. Adjustment of the parameters of the door with the help of a picture.



In order to adjust the 3D model to measured data, laser point clouds can be inserted within the 3D scene. However, the great amount of points makes it difficult to identify relevant elements and to adjust the parametric components. This is the reason why we decided to resort to the approach proposed by Boulaasal et al. [8] to complement our system. These researchers developed a method to automatically compute relevant straight lines and curves whereby they generate parametric models of the wall and its openings (see Figure 7). In our graphical user interface, the point cloud can be displayed or hidden; it is possible for the user to toggle between the two options. The user can visualize the lines and curves that enable the system to compute the parametric element (displayed in green in Figure 7). It is then possible to complete the 3D model of the building front by inserting the architectural details adorning the openings using our G.U.I. It is a matter of adding the elements dividing the opening (when it is subdivided) and also juxtaposing the upper, lower, right and left elements. While doing this, the dimensions of the constituent parts can be adjusted by making use of photographs or plans.

## 7. Examples

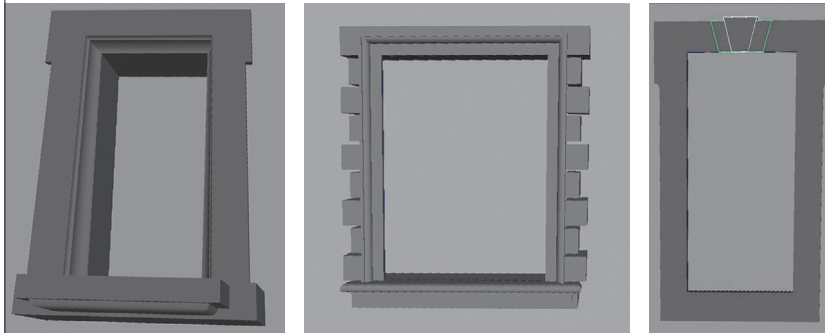
Figure 9 shows some examples of existing windows and their 3D model. Figure 10 shows some examples of 3D models of lintels.

## 8. Conclusion

Our experiments led to promising results. Up until now we have laid the foundation for a more complete system. We will now endeavour to improve both its flexibility and performance. On the whole, the major limitation of this approach is twofold: on the one hand, as regards to openings, the selected districts of Nancy and Montreal encompass a wide range of



Pictures of openings in Nancy and Montreal

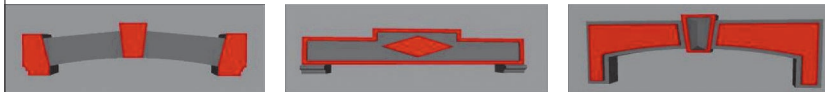


The corresponding 3D model generated with our parametric elements

◀ Figure 9. Example of stone elements defining the windows' outline.



Pictures of lintels in Nancy and Montreal

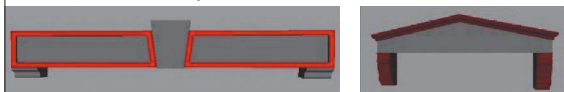


The corresponding 3D models generated with our parametric elements

◀ Figure 10. Examples of lintels with sub-elements: corbels, abutments, cornice, keys, etc.



Pictures of lintels in Nancy and Montreal



The corresponding 3D models generated with our parametric elements

configurations. Obviously, we will not be able to take into account the whole gamut of variations. On the other hand, we do not pretend to elaborate parametrical objects capable of generating 3D models representing architectural elements down to the last detail. Our system combines geometrical primitives (liable to undergo deformations later on) or creates surfaces from curves; the modelling process of sculptural elements (such as bas-relief) is beyond the scope of this paper. However, we believe that this approach could lead to the elaboration of an efficient tool.

The use of parametric objects leads to a significant saving in time by optimising the modelling process. Such a tool has proved to be useful within the framework of 3D virtual reconstruction projects dealing with rather homogeneous built environments.

## Acknowledgements

Great thanks are due to Hakim Boulaasal who extracted the data from the laser point clouds [8].

## References

1. Maya software: <http://usa.autodesk.com> [1-9-2009].
2. Photomodeler software: <http://www.photomodeler.com> [1-9-2009].
3. Debevec, P., Taylor, C. and Malik J., Modeling and Rendering Architecture from Photographs: A hybrid geometry and image based approach. In: *Proceedings of SIGGRAPH*, pp. 11–20, 1996.
4. El-Hakim, S. and Beraldin, J.A., Detailed 3D reconstruction of Monuments using Mutiple Techniques. In: *Proceedings of the international workshop for cultural heritage recording*, Corfu, pp. 58–64, 2002.
5. Mueller, P., Zeng, G., Wonka, P. and Van Gool, L., Image-based Procedural Modeling of Façades, ACM Transactions on Graphics, In: *Proceedings of SIGGRAPH*, volume 26, number 3, article number 85, pp. 1–9, 2007.
6. Trimble: <http://www.trimble.com> [1-9-2009].
7. Remondino, F., From point cloud to surface: the modeling and vizualisation problem. In: *Proceedings of ISPRS International workshop on vizualisation and animation of reality-based 3D models*, Tarasp-Vulpera, Switzerland, 2003.
8. Boulaassal, H., Landes, T. and Grussenmeyer, P., Automatic extraction of planar clusters and their contours on building façades recorded by terrestrial laser scanner. In: *Special issue of the International Journal of Architectural Computing IJAC*, Issue 07, Vol. 01, pp. 1–20, 2009.
9. Chevrier, C., Charbonneau, N., Grussenmeyer, P., Perrin, J.-P., Élaboration de nouvelles configurations à partir de cas existants du patrimoine bâti, In: *Actes du 3<sup>ième</sup> Séminaire de Conception Architecturale Numérique*, Bigon, J.-C., Halin, G. & Kubicki, S. éditeurs, pp. 111–122, 2009.
10. Chevrier, C. and Perrin, J.P., Generation of architectural parametric components. In: *Proceedings of CAAD future 2009*, Presses de l'Université de Montréal, Montréal, pp. 105–118, 2009.
11. De Luca, L., Véron, P. and Florenzano, M., A generic formalism for the semantic modeling and representation of architectural elements. In: *The Visual Computer*, Vol. 23, number 3, pp. 181–205, 2007.
12. Fuchs, A., *Outils numériques pour le relevé architectural et la restitution archéologique*, PhD thesis, Université Henri Poincaré, Nancy, 2006.

C. Chevrier and J.P. Perrin

UMR MAP 694, Equipe CRAI, ENSA-Nancy,  
2 rue Bastien Lepage, 54000 Nancy, France

(chevrier, perrin)[@crai.archi.fr](mailto:chevrier, perrin@crai.archi.fr)

N. Charbonneau and P. Grussenmeyer

UMR MAP 694, Equipe Page, INSA de  
Strasbourg, 24 Bld de la Victoire 67084  
Strasbourg Cedex, France

[nathalie.charbonneau@umontreal.ca](mailto:nathalie.charbonneau@umontreal.ca)

[pierre.grussenmeyer@insa-strasbourg.fr](mailto:pierre.grussenmeyer@insa-strasbourg.fr)