MaquetistaVirtual
Interactive Procedural Modeling

Luis Miguel Saraiva Lopes
IST / Technical University of Lisbon
Portugal, Lisbon
luismslopes@ist.utl.pt

ABSTRACT

The techniques used in existing modeling applications prove to be time consuming, generic and inappropriate for models that have similar components, such as buildings. This led to the creation of a new approach to generate models from a textual description, called procedural modeling. Few users are able to make the necessary descriptions, continuing the problem of the creation being too long, tiring and difficult to learn. From this emerges the motivation to create a new and expeditious procedural modeling environment that will match the user’s skills. We aim to provide an interactive, natural and better learning procedural modeling approach than current ones. We consider the models edition by interacting directly on its representation, adapting a wiimote to enable procedural operators. To visualize the model we propose a solution of active stereoscopy which allows the user to observe the virtual model in a table with a depth feeling, watching it as if it were a physical model. We also propose functionalities such as component duplication and a template library, which abstract the user from procedural concepts and speeds up edition. We performed an experimental evaluation which primary purpose was to check feasibility of the proposed approach. The results of this assessment highlight the user preference for our more immersive and interactive approach. We conclude that it is feasible to create an immersive modeling that allows for a more interactive, enjoyable and good learning procedural edition.

1 INTRODUCTION

For architects, the models are the best way to make a realistic representation of large-scale structures such as buildings. Recently, the introduction of computer-aided design (CAD) systems has enabled the spread of virtual models in architecture studios.

The techniques used in existing modeling applications allow the model creation by specifying each component individually, being too lengthy, generic and interactively inappropriate for models that have many similar components, such as architectural models. This motivated a new way of modeling, called procedural, that uses computer techniques to generate 3D models from a textual description, usually done through a series of rules that can being embedded in the algorithm, obtaining very realistic models and with less time spent on the process of creating models.

Currently, applications commonly used for three-dimensional modeling, such as AutoCad, 3DStudio Max or SolidWorks, are based on interaction via mouse and keyboard following the approach WIMP (window, icon, menu, pointing device), which is a time-consuming, tedious and not very swift modeling approach, limiting the expression of designer ideas and requiring a long and slow learning.

However, despite the inappropriate methods of interaction and little innovation, or applications such as iCube CAVE allow a better perception of 3D objects through immersive virtual reality environments where the user feels more involved in his virtual world.

It is therefore important to know what is needed to build a system with the appropriate user interfaces, in order to increase the comfort of users during the design process of three-dimensional models.

Keywords Procedural Modeling, Stereoscopic Visualization, Natural Interaction, Immersive Environment.
of buildings.

Thus, emerges motivation to create a new and expeditious modeling environment that will meet the user’s skills, improving the interaction techniques in procedural modeling.

In this paper we develop a modeling environment that integrates the immersive virtual environments with procedural modeling which allows a faster, more interactive and less boring edition of buildings models, with the aid of tools that speeds component changes.

We intend to provide a more interactive, natural and better learning procedural modeling approach than the typical approaches, through auxiliary tools that motivate an increase in user satisfaction with this modeling type.

Hence, we propose a solution that encompasses three different contexts, each with its valences and challenges: the creation of more natural and quicker user interfaces for modeling purposes; the development of features allow the user to abstract from procedural concepts, enabling a quicker editing of their models and, finally, visualization of models in an engaging and natural approach.

To visualize the model we provide an active stereoscopy solution that enables users to observe the virtual model at a table with a depth sense.

We also allow editing by interacting directly with the model, which is possible by following the movement of an interface with buttons that apply the modeling operators. As in the visualization, since there is no restriction of movement, the user can move freely in their environment to edit the template. With this solution error conditions are avoided due to reduced ambiguity in the interpretation command button, also we can change all the components of the model regardless of their size or location.

In order to validate the solution options, to verify the impact of the high-level editing features and to prove the feasibility approach and learning time of the proposed approach, we performed an experimental evaluation. For this evaluation we developed a prototype solution that would enable two interfaces, an immersive and non immersive another, so we could make a comparison between the two approaches.

The experimental evaluation consisted of three tasks to ten users, divided into two groups, each for a given approach, followed by a questionnaire.

From results analysis we highlight the fact that all users were able to finish the tasks within a similar time to the typical approach. We also highlight the user preference for a more immersive and interactive approach as the proposed one, and also the satisfaction gained from the modeling support tools.

From these results we retrieved the following main contributions:

1. An immersive approach that allow typical operations of procedural modeling, such as division, repetition and extrusion, to be completely abstract for the user and applied directly on the model.

2. Introduction on a procedural modeling environment for new editing tools for high level, as a template library, duplication of model components and a copy buffer.

3. It is feasible to create an immersive modeling environment that allows for a more interactive, enjoyable and good learning procedural edition.

2 RELATED WORK

To be able to make a consistent effort and a good theoretical base, we looked carefully to some developed systems in a few areas related with this subject. The analysis is divided into two main themes, the procedural modeling and virtual modeling.

Procedural modeling

With the advancement of modeling techniques, new approaches that aim to automate the modeling in order to reduce the time required to produce elaborate models that can be structured in a hierarchical manner and which may contain many repeated patterns. In this context, the first systems to quickly create very realistic models of plants, emerged, such as in Power et al. [13] and Bouden et al [1].

This type of modeling is called procedural and the main objective is the creation of large models through the specification of some rules or using algorithms.

In these systems it is usual to use grammars to describe the models that are generated procedurally. In the modeling of buildings and cities appeared the grammar of forms that are an adaptation of the L-system grammars, and are specifically used to generate geometric shapes.

However, the grammars have a very significant learning curve, especially for users who are not accustomed to concepts of programming, moreover,
the grammars are limited to symbols which they interpret, and you can’t perform actions that do not exist in the symbols grammar.

The city models have some characteristic that favor the use of procedural modeling for its construction, like the existence of several repeated patterns in their shape or the ability to structure the content in a hierarchical manner.

Up to 2001, systems that built cities procedurally relied on too much geographical information (height, terrain, ...) and social information (density, height map, ...). However, Parish and Mueller [12] showed that the CityEngine can model a complete city using less information, which is mostly used to generate the road network, but the buildings are generated through user-defined rules.

Wonka et al [15] reach the conclusion that is not enough to define the rules of the grammar to obtain buildings that are consistent, credible, varied and that meet certain criteria. To do this, they propose the Grammar of Division that are an extension of the Grammar of Forms, where the application of rules splits the shapes into smaller ones.

The Division Grammars have an important role in simplifying the specification of the rules. For example, suppose you want to specify the rules for constructing a façade with four windows. Then we split the original four symbols symbol "front", which in turn is divided in the center with the "window", with the top of the window to containing the symbol "adornment" and the terminal symbols "wall" at rest. The adornment is a set of other terminal symbols. The symbol "window" will in turn be divided into two terminal symbols called "sill" and "windows" which are two textures of stone and glass window respectively. Since there is no more terminal symbols, the derivation is finished (fig 1 esq.).

Mueller [10] returned to work on the modeling of buildings, but this time they created facades with a plausible resemblance with photographs of an actual building. One major difference of this system is the use of a 3D Objects library to insert low-level architectural elements. This cause the results have a higher visual quality without the need of more time modeling.

Several methods are available for editing: Segments manipulation in the graph; Handling vertices; Relocation of streets; edition of streets in layers. There is also the possibility of applying the tensor fields on a local level, allowing, for instance to create a park within a city with a typology of streets different than the general topology, including the corner of "local" to the streets "general". This approach has the disadvantage that the results are hardly controllable because of the modifications are made in an indirect way.

Lipp et al [6] introduced a system of editing and visual interaction in ways that grammars, inspired by the paradigm of providing direct control over indirect, makes possible the realization of a graphical editor for procedural modeling.

One of the barriers that existed was the difficulty of making local changes, for example, change the size of a single window [15]. Therefore, they propose a mechanism called "accurate locators" that allows clearly specify the instance in a hierarchical tree and so associate the variables to the locator, this means, a local modification.

They also have "semantic locators" who, as the name implies, allows the selection by the semantics of the elements (for instance, the entire facade, floor, column ...).

Another problem identified was the lack of persistence of the changes made to the grammar [1]. To resolve this problem, the couple locator + attributes are stored externally and will be called during the model generation.

The visual editor lets you edit the grammar through a toolbar with the commands of grammar (Split, Repeat, Split Component ...) and has a tree view of model rules which allows editing of its attributes. In addition, you can preview the model in 3D and edit it directly by manipulating the plans and the divisions with the grammar commands (fig. 2). This system was the first editor, with a fully procedural basis that allows a user to create and modify rules grammars without requiring any textual editing.
Virtual Modeling

Usually when we refer to a virtual reality environment, we mean one in which the user is totally immersed and is able to interact with a completely synthetic world. However the concept of virtual reality has been associated to other types of display which concepts of total immersion and fully synthetic does not apply [8].

Milgram et al. [9] introduce the notion of Reality-Virtuality Continuum where in the extremes we have the virtual environments on the right and left real environments. Virtual environments can be classified as non-immersive and immersive. The immersive environment is one where the user feels involved in the virtual world and whose actions they carry out in the real world have some kind of direct correspondence with the virtual world. In the non-immersive environment the user can distinguish perfectly the real and the virtual world by viewing the virtual world through a window, without have a direct correspondence between them.

In order to the user have a better understanding of the connection between the real and virtual world, it is usual to make a correspondence between a physical object and a virtual object. Systems that have a correspondence between physical objects and the virtual world objects, where the user interacts with the virtual object by manipulating the physical, are called tangible interfaces [2].

One type of tangible interfaces in augmented reality scenarios are the physical markers that represent objects in the virtual world possible to be handled according to the marker movement. To do this it is usual to use tracking techniques.

These mechanisms can also be used to capture the movement of the user's hands, working as tangible interfaces linking them to the virtual representations of the hands [3] or objects.

A recent approach is to use the Wiimote Nintendo command that serves as tangible interface. Schlömer et al. [14] achieve good results gesture recognition using this command. However, you can't make selections in a virtual environment because is not possible to determine the location in space of interaction of command being necessary to use markers and infrared sensors. In addition, the command also proved to be unsuitable for operations that require greater precision, such as selections and local modifications of the model.

3 SOLUTION

The developed solution offers a new and expeditious approach to editing models generated with procedural modeling techniques, that facilitates learning of the concepts of this type modeling and that abstracts the user to their specific conditions.

3.1 Overview

The approach is developed in three different contexts: Interaction - Creating new ways of interacting with the model; Modelling - The development of new mechanisms of change in abstract models, Visualization - The visualization of models in a more natural and more involving approach (fig. 3).

The interaction mode of the proposal deviates from the accomplished through traditional means such as mouse and keyboard and is composed by means wishing to engage the user in a modeling environment more interactive, providing a combination of interfaces in an immersive environment (3DU1).

These interfaces use editing mechanisms to procedural models, without being necessary to have the knowledge of the editing operators specificity. This change management model (GA) along with procedural generation (WG), form the context to modeling of this proposal, where is created and changed the internal representation of the model.

From this representation, graphic shapes are created to make up the visual feedback given to the user (OSG), which is constantly updated to resemble the internal representation that the model has in a given moment, in this manner is given a real-time feedback of the model editions.

To view the model and its editions, it is necessary to eliminate the barrier that has the monitor regarding to the visualization of the scene. Thus, the proposed approach allows the user a natural
movement around a virtual model as if it were a physical model. Once the user does not lose track of the real space around him, it manages to get involved with the modeling environment available.

3.2 Visualization

The graphical visualization of models has a major impact on users experience and influence the decisions they make for the remaining components of the solution, such as the way of interacting with them and, consequently, how they are edited.

We analyzed several visualization approaches and the most suitable viewing type is the semi-immersive virtual reality environment because it is possible to partially see the real world and, simultaneously, different angles from the virtual model. This allow us to avoid the use of a camera to film the scene, making it more cleaner and versatile.

To have a really effective stereoscopy effect in this scenario is necessary to adapt the visualization of the scene at the users head position and orientation. Due to the characteristics of our approach, the adopted technology needs to allow natural and unrestricted movement, besides having a good degree of accuracy for some modeling operations (eg. division).

For this, we use a tracking system that allows us to record and reconstruct the motion based on image analysis, generated by ten optical-electronic devices (video camera) and by reflective markers of infrared light (passive markers). There are no restrictions on the form or type of characteristic of the movements that can be analyzed, although the factors that limit the movement analysis will be imposed by the cameras and lighting used and the number and position of the markers.

We placed five markers on the glasses forming a virtual rigid body which represents only one position and orientation. These markers were placed asymmetrically to avoid interpretation problems of its orientation.

3.2.1 Feedback Mechanisms

To be able to achieve the aspirations of an easy and simple interaction, with reduced learning time and, consequentially, a better average duration of tasks, we still needed to provide the user with a visual perception with the consequences of their actions. To this end, the application gives the user some visual clues on the actions that is to be held at a given moment, especially in the selection and editing forms.

Selection

There are three different ways to select components: the main, the auxiliar and the functional selections. Each with a different purpose and with an appropriate visual feedback.

The most important way to select components is the main selection mechanism which uses an hierarchical selection. With this selection type, we can apply all forms of transformation, and it is the only one on which we can apply two major operators, division and repetition. All selected components are highlighted with a red colour (fig. 4).

However, this selection is not enough because sometimes we need two components to make a change. For example, the duplication of a component from the middle of the hierarchy to another component, also in the middle of the hierarchy, needs to have two components simultaneously selected, one as an origin and the another as the destination.

To meet to such situations is provided an auxiliar selection that also uses the mechanism of hierarchical selection. This selection can only be used if there is somewhere another component selected with the main selection. Since the red highlight is already taken, for this type of selection, we highlight components with a yellow color.

Even with these two types of selection, it would not be possible to make simultaneous changes to a set of components. For this reason, it is available a semantic selection that selects the components of the model that give the same semantic perception to the user, such as every windows of a floor(fig. 4).
Editing Templates

We also developed forms of visual feedback to editing operators that can be applied in the models, most notably for resizing, splitting, and repetition.

In this approach the user can perform a division, indicating the cut at any point of the selected component, in a vertical or horizontal direction. When making the cutting motion with the command, it appears a black line along the cutting being performed.

Similarly to the division, in repetition user makes a movement towards the direction it wants to repeat the component that is selected. By performing this gesture, appears a blue line parallel to the axis movement and position in which that will occur.

There are also other operators and tools possible to use and that require some kind of feedback, even if not specifically dedicated. Therefore, at the end of any attempted application of an operator, its success or failure is indicated by the icon next to the pointer.

The application of colors and textures operators will not have a dedicated form of feedback. They were featured at a menu that eases access to grammar tools.

3.3 Interaction

It is intended that the user interacts with the MaquetistaVirtual in a natural and quicker way, with minimal errors, thus, as noted by Young et al. [11], it is essential that the interaction is fault-tolerant unconsciously held by the user. To this end, the interfaces used must have a good interaction completeness and interpret the commands with lower ambiguity.

The proposal interaction approach should be easy to learn, therefore, should only be available at most ten gestures, so the user doesn’t need to make an effort to memorize [17].

Besides the ease of learning, it is intended that the approach has enough precision to be able to change low-level components like a window, door or balcony [7], that is, sufficiently specific to make local manipulations in the model.

Model Edition

With the aim of directly editing the virtual model, it was necessary to establish an interactive method that would apply, quickly and naturally, different editing operations on the model or edit attributes such as color or texture.

We choose to perform an approach that mixes an interface of a pointer with the approach of wiimote. Apart from both approaches to be identical to the interfaces that users are accustomed to use as drawing tools (pencils and pens), the approaches are physically complementary, since the wiimote command already has the necessary buttons and if we trace its movement, they float the internal sensors and it is possible to increase the command accuracy while editing the model. However, when we decide for this solution we require that the space surrounding the user in the interaction environment is prepared to track the movement of the command.

The developed solution has two different components: the tracking of movements and the suitability of command buttons to the developed editing features. In order to obtain a more precise position and orientation markers were added to the wiimote, allowing the optical tracking of the command, like the stereophics glasses were to enhanced a natural visualization (fig. 5).

To allow the user to see the exact spot he was pointing at the model, we create a virtual extension of the command, like a physical pointer. With this solution we have increased the precision while handling the wiimote, quickly identifying any model component, regardless of size or distance from which it is located.

Model Space Manipulation

As the physical models, where you can manipulate the model by positioning it in order to be observed from a better perspective, our solution allows the user to manipulate the virtual model. To this end, it was necessary to establish a solution that enable the user to manipulate the model in order to look at it from different perspectives, taking into account that it should be performed anywhere on the table, in a natural way, and with the non dominant hand since dominant one will be busy with the wiimote to apply operations on the model.
In the adopted solution, the user places his fingers on the multi-touch surface to indicate the desired type of manipulation by the number of fingers placed. The user can switch the number of fingers to the desired type of manipulation while not in the same place for a second, after which it will set the type of manipulation that is being interpreted. Moving the fingers to manipulate the model space.

### 3.4 Modeling Operators

Choosing procedural modeling techniques for the generation of models was due to the ease of dealing with repeated patterns, such as buildings. Thus, the proposed system allows to specify procedural models through a CGA Shape grammar, similar to that proposed by Pascal and Mueller in CityEngine [12].

We also provide a set of features to perform basic editing operations, which make procedural concepts transparent to the user and allow the immediate change of architectural components in the model. It is possible to perform operations that give volume to a shape (extrusions), change shape geometry (geometric transformations), divide into two or more shape (division), repeat a shape along a space (repeat) or as simple as changing its color or texture.

Beyond these basic editing features there are available some tools that aim to facilitate and accelerate model editing. The purpose of some tools, like copy-paste or duplicate, is very similar to others widely used in modeling applications or even in simple editor applications, having recognized benefits to facilitate the editing process.

While the application of simple operators changes a component at time, these auxiliary tools speed up model editing by applying a set of operators at once. Common to all these tools is the process of copying a description associated with a component that is selected. This information is stored for later use or to be applied in any other component of the hierarchy by removing all of which below.

Another important tool added emerged in response to an obvious lack in other procedural approaches (eg. in Yi et al. [16]) that was a repository of recurrently used architectural component models (windows, doors, moldings, etc..), and that can be swiftly applied in the models by users. Thus it was introduced a library of pre-defined architectural components corresponding to several architectural styles, to be used either in the specification of initial models or later while editing the generated model.

With all these features, it is possible to create complete and detailed models without resorting to textual descriptions, menus or any representation of the model internal hierarchy.

### 4 EVALUATION

To evaluate the prototype in terms of usability, a study was carried out with ten users (fig. 6), comparing the results of implementation and satisfaction with a method of editing that is not immersive, similar to the system, a procedural creation of buildings proposed by Lipp et al [6]. We wanted see the level of learning, usability and satisfaction that our system requires and the extent to which users can perform faster models using MaquetistaVirtual.

We recall that the ultimate goal of this work is to provide an approach to editing procedural models, faster, more interactive and less boring than the approaches of this type of scripting typical modeling. To achieve this goal, we developed a solution that combines stereoscopic visualization on a table where operations are made directly in the model through a command and a multi-touch surface.

So, it becomes important to analyze all of the procedural features developed, drawing conclusions as to the satisfaction of users in relation to these tools and analyzing their contribution to the performance of the approach.

Since a major focus of the approach is on user interfaces, the objective of this evaluation is to draw conclusions about the naturalness and ease of use of the developed interfaces, as well as demonstrate the feasibility of modeling tasks in a semi-immersive environment. We intend therefore to
draw conclusions about how far and how it is that this approach is able to replace other solutions interactive procedural modeling, and draw some conclusions about the requirement of the learning process so that a user feels comfortable in using the solution proposal. Easily verified that some of these targets imply a comparative analysis with other procedural modeling approaches, so we decided to make a comparison with an approach that is closer to that shown by Lipp et al.[6], where the user can interact directly to the model through its display on a monitor, using the typical mouse and keyboard interfaces.

Figure 6: Two test users interacting with MaquetaVirtual

With the same metrics in the two approaches we hope to better demonstrate the validity of the developed solution, comparing the gains achieved with our approach, which aims to be more interactive and natural and therefore faster. The tasks to accomplish the experimental evaluation cover the different features of the system and are organized in a gradual increase difficulty.

Figure 7: Average execution times for each test group

The motivation for each task and the goals that users should achieve were:

1. In the first task, we expected that the user had a first contact with the editing features, and therefore the objective of the task was to use the simplest editing operation and an auxiliary tool that our solution provides, using the most simple type of selection.

2. In this task, we wanted the user to realize the practical effect of effect of procedural concepts drawing conclusions about its benefits, so the objective of the task was to make two editing operations, one on the model and the other using semantic selection, allowing the user to understand procedural concepts.

3. In the last task, we wanted to analyse the proposed approach with the construction of a model from its root and also analyse the benefits of direct interaction with the model and auxiliary tools. Thus the task objective was to build a model, using a more advanced operator and the library of templates to build parts of it.

We collected each time the user took to accomplish the tasks identified in order to analyze the use of the developed interaction methods. The users started the tests in 2D or 3D environment. So, the results were divided into four groups: users that started with 2D vs the users that started in 3D environment and the consequent results in each environment.

In the figure 7 we can see the average running time of each group. We observe that the execution time of tasks in the first trial, when they experience the first approach to 2D or 3D, is very close, not revealing a very significant difference. This indicates that there is not a big difference in performance of tasks when they have their first contact with the system.

Although the execution times of tasks in the 3D approach could be worse than expected, it is not possible to conclude that a typical procedural modeling approach is faster than the presented interactive and natural approach. It is not possible to draw this conclusion due to the fact that the comparison was made between the same set of features, where the approaches were different only in the interaction methods. Thus, the 2D approach does not really represent a typical procedural modeling system, allowing, for example, direct application of operators on the model, the absent of hierarchical graphs for the selection and availability of auxiliary tools like the models library or the duplication.

For the same reason, it is not possible to draw clear conclusions about the influence of the high-level editing procedural features in the performance of users, however it is possible to make a simple comparison with the experimental data from a study by Lipp et al.[6], which was the first
work to develop a fully procedural editor that enables a user to create and modify rules grammars without requiring any textual edition.

We can unequivocally state that users prefer the proposed approach in all measured parameters, which indicates the success of a more natural and engaging approach in the perception of users, verified by the good acceptance of particular points as the modeling tools.

Thus we can state that the most important conclusion to be drawn from these results is that you can provide an immersive modeling environment that allows making changes to a procedural model, in a more interactive, enjoyable and natural way to the user. This conclusion is mainly caused by the simple fact that users were able to finish the tasks in an average time similar to the 2D approach, but with far greater enthusiasm during its realization.

5 Conclusions

In this chapter we address the final conclusions with talking about the contributions of our work and points that might be taken in the future.

5.1 Contributions

From the obtained results in evaluation, we conclude that the proposal approach consists in a capable solution to realize editions in a procedimental model through an immersive environment interface, thus resulting in several contributions.

We realize that users without any knowledge in procedural concepts, were able to use the available operators and claimed to recognize characteristics of these modelation type. So, we contribute with an approach that allow us to realize typical procedural modeling operations, like division, repetition and extrusion, in a complete abstract way and with direct manipulation over the model, without the explicit use of scripting to the definition of rules or menus.

Another contribution was the introduction of new high-level edition tools, like the models library, the duplication and the buffer copy, in the procedural modeling environments. These tools allow us to achieve better times in the tasks realized and also acceptance by the users.

The last but not the least, through the evaluation, we conclude that is possible to build and edit procedural models with a multi-modal environment that contains a stereoscopy visualization, interaction though the wiimode and with a multi-touch surface.

So, we may state that the biggest contribution of our work is the conclusion that is possible to realize an immersive modeling environment which allow us to edit a procedural model, in a most interactive, natural, enjoyable way and also with a good learning.

5.2 Future Work

Throughout the investigation work, we identified some ideas that can be exploited in future work.

In the experimental evaluation (and its questionnaires), it would be useful to develop an undo tool, which allows the return of the model to a former state. The mechanisms used in this work can be used to provide this feature. Another feature, consists in the evolution of the selection methods, allowing to create exceptions in components, like the ones used in OCTOR[4] Jang e Rossignac work.

Regarding modeling environments, we discovered, also through the evaluations tests, that the wiimode is not a good approach because it contains a lot of buttons and consequently actions that turn out to be hard to internalize by novice users. So, we suggest a more profound study to discover a better way to interact with the hand on the model, whether using the tracking methods described in this work or the depth sensors.

Another attractive approach is the use of tangible interfaces to represent components of the procedural model. This results with a combination between the proposed approach and the one develop by Jota e Benko[5].

References


[3] J.M.S. Dias, P. Santos, and R. Bastos. Gesturing with tangible interfaces for mixed real-


