VirSculpt: a virtual sculpting environment

Romain Raffin, Gilles Gesquière, Eric Remy, Sébastien Thon

LSIS Laboratory
Marseilles, University of Provence, France
{Romain.Raffin, Gilles.Gesquiere, Eric.Remy, Sebastien.Thon}@up.univ-mrs.fr

Abstract

We describe a virtual sculpture environment suited to the needs of artists. In this approach, the user is immersed in a virtual environment by the use of haptic devices and immersive display. The project is focused around four main research topics: the sculpture model, the tools and their actions on the model, the display of the sculpture, and the interface between the artist and its virtual work. First results are presented and future works are discussed.

Keywords: *Virtual sculpture, volume data, octree, haptic devices.*

1. INTRODUCTION

Our project of virtual sculpture started with meetings of artists interested in the widening of possibilities offered by a digital creation framework. Most of them confess their difficulty to use computers during the creation process: the computer implies a conformation to peripherals, hardware and software.

In classical modeling software, the user faces several projection views like in technical drawing, but which are far from natural vision. Moreover, he or she has to interact with mouse and keyboard which requires a practice that traditional artists seldom have.

Our goal is to obtain an immersive sculpture environment that gives a convincing impression to the artist, adequately precise and fast to control the sculpting process. The classical input devices (keyboard and mouse) are replaced by a force feedback haptic device which reaction and prehension are more similar to classical sculpting tools. The visualization will be done with a head mounted display to immerse the artist into virtual reality and to provide a transportable solution.

Several virtual sculpting projects have been developed using surface representation, such as McDonnell's tri-parametric Bsplines surfaces [1]. This method is not as intuitive as real sculpting due to the control point structure.

A volume representation based on CSG can be used, but the increasing number of operations made during sculpting process implies a growing computational time. Such a model has been created by Mizuno et al. to sculpt virtual wood with an ellipsoidal tool as virtual chisel [2]. Its drawback is ray traced visualization, which is viewpoint dependent. The two usual CSG operations (union, intersection) linked to a tool could be improved with R-functions [3].

Another volumetric method has been implemented by Galyean and Hughes [4]; their model is defined with a uniform discrete spatial enumeration, however a uniform grid becomes expensive when precise details are needed. They also propose basic operations like addition or subtraction, and several tool definitions (heat gun, sand paper or color modifier).

Hierarchical representation decrease the memory cost of the structure. Bærentzen chooses an octree which permits to work on higher grid resolution than uniform enumerations [5]. The subdivision level is fixed and unfortunately prevents highly detailed objects. The only tool shape is a sphere, and the only operations are CSG (addition and subtraction) or surface filter "spray". The same problem of static ray-casting visualization suppresses visual interactivity. Ferley also works on a hierarchical organization of voxel: a cell can be divided in 27 ones [6]. It permits to reach a high level of detail. The tool is defined by an ellipsoid, which is artistically very restrictive.

This last method can be compared to [7], which uses distance fields in a discrete space octree. Operations are CSG on the object or direct triangle face modifications, but the Euclidean distance field used instead of density raises discontinuity problems and increases update computation time.

We think that a combination of a volumetric coding and Brep is interesting. On one hand, the volumetric coding is well adapted to express sculpture operations, and moreover, a hierarchical coding (like an octree) is more efficient than a regular one. On the other hand, a Brep (deduced from the volumetric coding) enables a fast display of the shape and the interactive modification of the viewpoint.

We denoted that the main difficulty is to provide adapted tools to the artists. We can not imagine a traditional sculptor interacting with unrealistic tools like an ellipsoid or mathematical functions.

The remainder of the paper is organized as follows: in section 2, we present our project in four parts: the coding of sculpture model, the sculpting tools, the display method and finally the user interface. Some first results are presented in section 3. We conclude and expose some future works in section 4.

2. VIRTUAL SCULPTURE PROJECT

2.1 Model

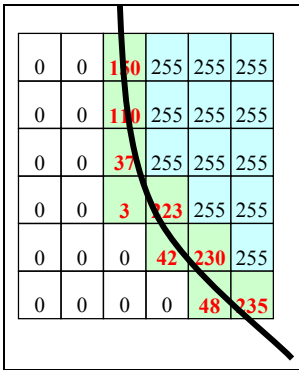
2.1.1 Data structure

Using a model based on a uniform spatial enumeration implies many difficulties to obtain a real time sculpting environment [4][6]. We propose in this paper to use a multi resolution approach based on an octree, like in [5] to lower memory cost. The sculpture is embedded in a volume recursively divided until the subdivided volume is empty or full of matter, or the fixed level of subdivision is reached. This level is initialized by the user but may be modified at runtime. If an octree node is empty or full we will name it a *leaf node*.

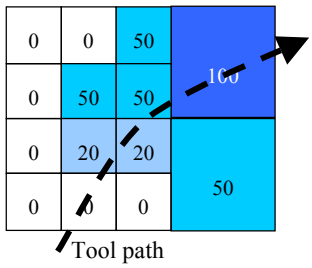
Each leaf node contains a density value coded in a byte (from 0 for an empty leaf node to 255 for a full one). An analogy may be

done with implicit objects. Every value under a previously defined threshold is not visible (see figure 1.a). These values are used to compute the interaction between a tool and the sculpted object. Another benefit is the possibility to avoid aliasing problems which occur when a binary approach is chosen as we will see in section 2.3. The tools which are described below will modify these values.

We choose to add other parameters to imitate physical behavior. For instance, we add a hardness parameter in each leaf node (see figure 1.b). This new parameter will simulate the ability to penetrate a sculpture with tools. If this parameter is low, the object will be malleable like clay. The position and the orientation of the tool with respect to the object are also important. We obtain indications on orientation by computing the hardness gradient field on each leaf node of the octree. The gradient is computed with the difference of hardness value in the tool direction. The bigger the hardness gradient value, the more difficult it is to enter the object using a tool with small hardness values.



a. The object is defined by the density upper than 150.



b. Array of hardness values used to compute the gradient.

Figure 1 Density and Hardness factors are stored in the structure.

2.1.2 *Primitives*

Basic objects, such as sphere, cube, cylinder, are proposed in a library. Moreover, we want to give to artists a maximum of freeness in the creation process, by enabling him to use any kind of object to initiate his work. For instance, the user can load polygonal objects made in other products like modeler software, which are then discretized using a ray casting method presented in [8].

2.2 Tools

Since we have defined the initial matter for the sculpture, we will show in this section the definition of a tool and the method to interact with objects.

2.2.1 *Sculpting tools*

Structures

We see that whatever the representation of an object is (Brep, implicit or voxel), it can be imported to be used in the sculpting environment; the same stands for the tools. Every kind of object described with voxels can be used as is (coming from an object library or made with another sculpture).

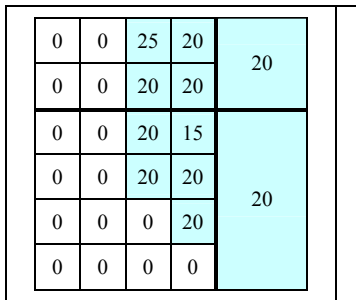
If the tool is represented by an implicit function, it can be efficient to keep the mathematical expression to compute distances or interiority, and polygonize the tool to improve its visualization.

Finally, a tool imported from Brep can be transformed into a voxel representation to be used directly in the virtual environment.

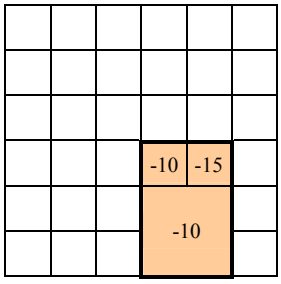
Operations

As the object and tools are embedded in a discrete space, using octrees, we need to set operations between their constituting nodes. Each node contains information on density, hardness or any other desired material property. The sculpting act consists in modifying these values to obtain addition, suppression or displacement of the matter.

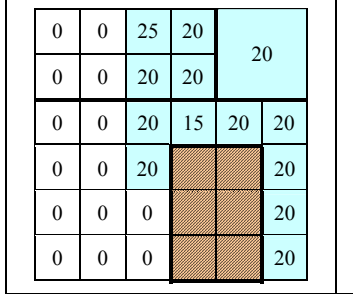
As the displacement of the tool is continuous, we perform the geometric transformation (translation, rotation, scaling) of the tool using a trilinear interpolation of its voxel values. The trilinear interpolation is used to reduce aliasing during resampling. We thus obtain an expression of the tool at the corresponding discrete location in the octree of the sculpture. After this matching phase, the chosen operation is applied to the values of the corresponding nodes. For example, as illustrated by figure 2, the initial object, in an octree representation (a) is hit by a tool (b) which may suppress matter. After the matching phase (c), the computation of the new values of the object is made (d). The octree structure of the sculpture can be modified (subdividing or gathering nodes) by the tool operations (cf. figure 2.c).



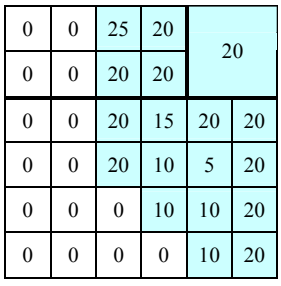
a. Initial object



b. Tool



c. Collision detection, modification of the octree describing the object



d. Resulting object

Figure 2 Subtracting matter to an object with a tool.

2.2.2 *Material tools*

The matter model has been constructed with values linked to nodes. If we want to sculpt heterogeneous object, we had to define several ways to obtain the initial matter. This principle is

identical to the definition of the tool and can be seen as an improvement on the preceding method.

For example, the artist wants to work with an egg as initial matter. He has to define an ellipsoidal object, with a solid and crispy surface, a flabby interior and a dense rubber centre. All of this can be done by modifying nodes values of the matter. We can imagine a spray tool, acting in the object or out, fixing values of matter characteristics.

2.2.3 Surface tools

As well as sculpting and material tools, our sculpting environment will also offer tools that modify the appearance of the surface of the object. These surface tools will act on the color of the surface.

These tools are similar to classical tools in 2D graphical applications, such as brushes. The color of the surface is modified by changing the color stored in the voxel influenced by the tool.

2.3 Display

Although the object is stored in memory as an octree for the sculpting process, it is displayed with OpenGL as a set of triangles. This triangulated surface is obtained from a marching cubes algorithm [9] applied to the octree structure. Thus, we obtain a smooth surface instead of a set of boxes.

In the case of a binary coding of the object, the marching cubes algorithm would lead to a rather angular rendering: the surface would always run exactly in the middle of two inside and outside voxels, thus generating angles multiple of 45°. But as we have density values in each node, we use them during the marching cubes to obtain a smoother set of triangles by an interpolation of the nodes density values.

In order to improve computation time, the marching cubes should be computed on the whole object only once, as a preprocessing step before sculpting. Then, during the sculpting process, the marching cubes algorithm would only be applied to the voxels modified by the sculpting tools, thus computing only the new triangles of the surface locally altered by the tool as in [4].

2.4 Interface

2.4.1 Hardware

Haptic devices

Our research prototype started using classical input devices such as mouse and keyboard, but we aim at a more natural way of interacting with the model. The first step towards natural control on the sculpting operations has been achieved by the addition of a Spaceball5000[1] device (cf. figure 3.a). This kind of input device can detect any combination of 6 degrees of freedom: 3 rotations and 3 translations. It is well suited to handle the position of the sculpted model as it enables to turn and move the sculpture like if it was in the sculptor's hand. The artist uses his left hand to turn or push the ball, and the corresponding movement is applied.

We also tried to use this device to control sculpting operations, i.e. to control the tool instead of the sculpted model, but we did find this method rather unsatisfactory for two main reasons:

- First, conversely to the movement of the whole sculpture, moving the sculpting tool (which may be very small) did require a high level of accuracy. This accuracy is very hardly obtained by the artist since the translations are given to the system by pressing the ball in a particular direction, and it requires a lot of skill to combine both rotational and directional pressures to perform in a single try the desired interaction.
- Then this device does not really move in the hand of the artist, nor can it give any feedback to him when the tool hits the model. Therefore, the sculptor can only rely on his visual perception of the operation to control the position of his tool.

To address these two issues, we plan to use a Phantom[2] device (cf. figure 3.b) in addition to the Spaceball. The artist will have the stylus of the Phantom in his right hand, and will be able to move it the same way in both the real and virtual space. Therefore, he or she will have the direct sensation of the location of his tool. Moreover, as this is a force feedback device, the artist will also be able to feel the contact of his tool with the surface of the sculpture.

a. Spaceball 5000.

b. Phantom 1.5.

Figure 3 Haptic devices.

Besides, alternative choices are also possible, such as the use of data gloves or a gesture recognition system but will not give the feedback we desire.

Display devices

For the moment, our prototype uses a rather simple dual screen display equipped with classical monitors or video projectors, which give a rather good viewing surface but no perception of depth. This flaw may be easily corrected by improving the display system using LCD shutter glasses.

In the future, we plan to use virtual reality rendering devices such as Head Mounted Displays to immerse the artist in a purely virtual sculpting environment or, using augmented reality techniques, to place him in a virtual environment embedded in a real sculpting studio.

Contrary to static sculpture environments, such as those based on Workbench or CAVE systems, the described project will be transportable, thus allowing the artist to perform anywhere, even in public places.

2.4.2 Software

The software is designed to permit the artist to chose a primitive shape to start his work (a sphere, a cube, a polygonal 3D object discretized using [8], etc), to choose his tool in a different set of primitives (both discrete and continuous). This tool acts like the set of brushes in a classical 2D paint program. But the system is also designed to permit the artist to use one of his previous

[1] See http://www.3dconnexion.com/.

[2] See http://www.sensable.com/.

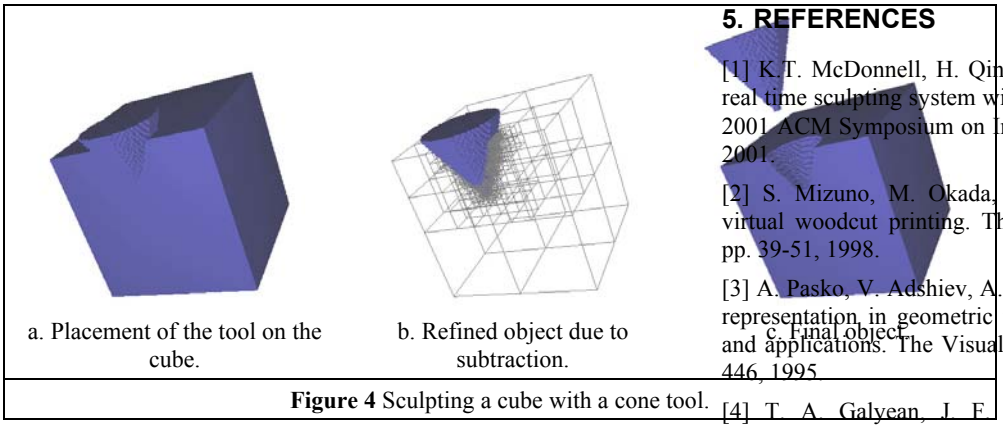
sculptures to be used as a new tool, widening the possibilities for complex designs.

The system is also designed to ease the choice of material properties (color, shininess, texture, rigidity, etc.), and then to "paint" them both on the surface and in the depth of the model. Some of these tools are not simple "painting" operations, but are also capable of adding or subtracting matter to or from the model (like pasting or carving).

3. RESULTS

Until now, we have mainly focused on the data structure definition for both object and tools, and interactions between them.

Both sculpture and tools have been coded as octrees. Interactions are controlled using three devices (Spaceball, keyboard and mouse). The transformations of the view of the sculpture (rotation and translation of the viewpoint around it) are controlled with the Spaceball in the left hand of the artist. The mouse in the right hand gives control on the tool (translation, rotation, scaling). Sculpting operations such as addition and subtraction between object and tools have been defined (see figure 4).

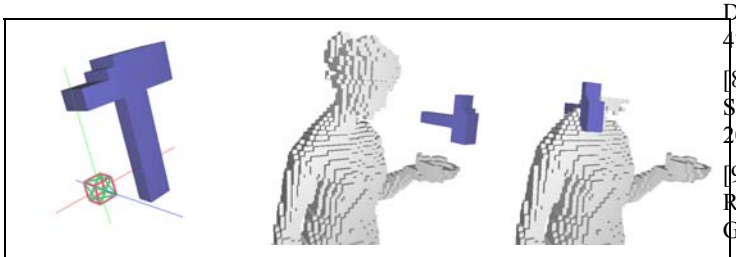


a. Placement of the tool on the cube. b. Refined object due to subtraction. c. Final object.

Figure 4 Sculpting a cube with a cone tool.

Tools of arbitrary shape can be used (Figure 5). Using a sculpture and a tool both coded as octrees leads to many problems in the management of their interactions. Particularly, establishing a correspondence between the tool at a given position and the sculpture is too time consuming due to the trilinear interpolation.

Currently, we have mostly worked on the modeling phase; the visualization one is in progress. We are developing an incremental marching cube, but this raises some problems to join triangles computed with cubes of different edges.



a b c

Figure 5 Sculpting an object with a hammer shape (a) then used as a tool to sculpt a voxelized Brep object (b,c).

4. CONCLUSION AND FUTURE WORKS

We have presented the framework of our virtual sculpture project. Our main objective is to provide an immersive virtual environment to artists, without the constraints of traditional computer devices.

This is an ongoing project. We focus our research on the four points presented in this paper: the model, the tools, the display and the interface.

Considering the model, we plan to add more parameters to the voxels in order to simulate more physical behavior.

We plan to propose more tools to the user, enabling more actions such as deformations or melting. A great challenge is also to optimize the computations in order to offer real-time interaction to the user.

As the purpose of our project is to provide a tool for artists, the quality of the display is also an important point to fulfill.

The interface has to be the more intuitive to use, in order to easily manipulate objects and tools, but also to easily select the functionalities of the environment.

5. REFERENCES

[1] K.T. McDonnell, H. Qin, R.A. Wlodarczyk. Virtual Clay: A real time sculpting system with haptic toolkits. Proceedings of the 2001 ACM Symposium on Interactive 3D Graphics, pp. 179-190, 2001.

[2] S. Mizuno, M. Okada, J. Toriwaki. Virtual sculpting and virtual woodcut printing. The Visual Computer, Volume 14(2), pp. 39-51, 1998.

[3] A. Pasko, V. Adshiev, A. Sourin, and V. Savchenko. Function representation in geometric modeling: concepts, implementation and applications. The Visual Computer, Volume 11(8), pp. 429–446, 1995.

[4] T. A. Galyean, J. F. Hughes. Sculpting: An Interactive Volumetric Modeling Technique. Computer Graphics, Volume 25, Number 4, pp. 268-274, July 1991.

[5] A. Bærentzen. Octree-based volume sculpting. *Presented at IEEE Visualization '98*, Late Breaking Hot Topics Proceedings, IEEE Computer Society Press, 1998, pp. 9-12.

[6] E. Ferley, M.-P. Cani, and J.-D. Gascuel. Practical volumetric sculpting. The Visual Computer, Volume 16(8), pp. 469–480, December 2000. A preliminary version of this paper appeared in Implicit Surfaces'99, Bordeaux, France, Sept. 1999.

[7] R.N. Perry, S.F. Frisken. Kizamu: A System for Sculpting Digital Characters, ACM SIGGRAPH, ISBN: 1-58113-374-X, pp. 47-56, August 2001.

[8] S. Thon, G. Gesquière, R. Raffin. A Low Cost Antialiased Space Filled Voxelization Of Polygonal Objects. GraphiCon 2004. September 2004.

[9] W. E. Lorensen, H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm, Computer Graphics, vol. 21, no. 4, July 1987, pp. 163-169.