

# Organic Shape Modeling through Haptic Devices

Renaud Ott<sup>1</sup>, Daniel Thalmann<sup>2</sup> and Frédéric Vexo<sup>3</sup>

<sup>1</sup>École Polytechnique Fédérale de Lausanne, [renaud.ott@epfl.ch](mailto:renaud.ott@epfl.ch)

<sup>2</sup>École Polytechnique Fédérale de Lausanne, [daniel.thalmann@epfl.ch](mailto:daniel.thalmann@epfl.ch)

<sup>3</sup>École Polytechnique Fédérale de Lausanne, [frederic.vexo@epfl.ch](mailto:frederic.vexo@epfl.ch)

## ABSTRACT

This paper presents a sketching system for 3D organic shapes modeling and animation using virtual reality devices. On the hardware side, it is based on the Haptic Workstation™ which conveys force feedback on the user arms (upper body limbs), and a Head Mounted Display to present the 3D generated images. On the software side, we use implicit surfaces modelling techniques such as metaballs. In fact, Designers feel well comfortable with this kind of primitives due to their ability in the organic shapes creation such as virtual humans. The proposed system provides an efficient alternative to produce advanced 3D shapes sketching.

**Keywords:** 3D Organic Shape modeling, Skeletal animation, Sketching, Two hands based haptic Interaction, Metaballs, Virtual Reality interfaces.

## 1. INTRODUCTION

Today, the 3D modeling and animation sketching is becoming one of the most important activity in the production of virtual reality application, video games production, and, special effect in cinema industry. 3D designers or CAD engineers should follow a long pipeline, starting from the sketching to the final rendering or the object creation. In [21] the authors show how this first step of design (sketching) is the most expensive of the creation process.

In this paper, we present a system, software and hardware, that speedups this first step of 3D design process. It is based on haptic devices which seem to us more efficient and natural to use than the traditional keyboard/mouse approach, because it allows to design with an interaction paradigm closer to the real one (e.g. clay modeling).

This paper presents, in the first part, a Computer Aided Design state of the art and the observations we could get from it. That will lead us to present a new concept; aim of research, Metaballs based Organic Shape modeling and animation with the help of Virtual Reality devices. In the second part, we will describe in detail its implementation. Finally, we will conclude this article with an evaluation of the proposed system to state advantages and drawbacks.

## 2. HUMAN COMPUTER INTERFACE FOR SHAPE ANIMATION AND MODELLING

First of all, we will study the usual 3D design and animation process used by professionals. Afterwards, we will see how it could be optimized with the help of haptic devices, especially when the force feedback can be applied on both arms and hands simultaneously.

During the creation of the 3D shape, some steps could be identified. The first one is always the same, whatever the “type” of production we would like to target: the prototyping. In this phase, most of the creators involved in the development process give their opinion on the 3D shapes and settle the problems inherent to the design decisions. Generally, this leads to the creation of a hand sketching with pencil and paper [9] or a clay based mock-up. We could add to this discussion the work of Alexe et al., Perry and al. which allow the creation of this first “prototype” electronically [1] [26]. These two last solutions have the strong advantage to do it quicker and allow mistake correction. The second step is the geometrical modelling phase which contains some refining. The third step is dedicated to the texture design and mapping. Finally, the designer could animate the modeled shape. As mentioned in [18], these phases or steps could be done in a different order according to the designer preferred methodology.

In this paper, we focus on the beginning of the process: the sketching. Lipson et al. in [20] precise, indeed, that an efficient and intuitive interface during the first step could save not only time, but also help the shape refinement necessary afterwards. However, Tovey et al. have shown the difficulty to create this type of system generic enough to be used in most industrial areas [32]. Based on their conclusions, we focus our system not on a generic purpose but dedicated to the virtual humans creation.

In fact, our research in the Virtual Reality Laboratory of Ecole Polytechnique Fédérale de Lausanne drives us to design very often virtual humans [30], [31], [23]. Despite the fact, that we have the habit to deal with commercial products like 3DSMax [10] and Maya [2] for sketching mesh and animation, we would like, in this work, simplify the life of designers. In [8], we had already proposed a solution based on a Cyberglove™ used to do generic mesh deformations for the creation of virtual human faces. The user was able to deform the mesh by virtual pressure on the vertices. This method is closed to the simulation of standard clay modeling when the user creates an object by deforming a piece of clay with the help of his fingers. This interaction paradigm is based of the theory of deformable meshes called Free-Form Deformation (FFD) [3] [7] [13] [14]. However, this technique has shown very quickly some strong drawbacks.

The first one is the lack of force feedback which creates a break in presence and reduces a lot the immersion of the user in his task. Nevertheless, FFD techniques could be linked without difficulties to haptic devices. It allows direct interaction with the vertices of the mesh. For example, we could mention in this area the work of Foskey et al. [11], Mc Donnel et al. [24] or Kuo Cheng et al. [17]: they all present a 3D mesh deformation system based on FreeForm™ [28] and the PHANToM [27], a haptic device that creates force feedback only on a single finger. This leads us to the next drawback.

The second one is related to the fact that interacting only with one hand or, even worst, with one finger is neither natural nor intuitive [5]. This implies indeed to switch to a second mode of interaction where the user is able to rotate or translate the 3D model. Foskey et al., McDonnel et al. “have solved” this issue by moving the PHANToM on a virtual button of the interface constraining the designer to lost the contact with its model. Moreover, working only with one hand on a soft object could lead to uncomfortable situations. In a general manner working with two hands seem more natural and intuitive, thus our interface should also takes advantage of the two hands.

As third remark, using only Free-Forms Deformation to model a virtual human from scratch is very time consuming. In fact, the FFD technique is more efficient in the refining process and not in sketching phase, because it needs a base shape to start deforming. This is the reason why we have studied the use of implicit surfaces [4] as an alternative solution. This allows with few primitives to approximate very complex shapes [29]. Moreover, they are very well adapted to represent organic shapes such as virtual humans or virtual animals [22]. A subset of implicit surfaces called metaballs [6] was becoming popular in computer graphics, because it could produce novel kind of shapes. However, this technique is not without constraint: we want to create a “wysiwyg” interface (“What you see is what you get”: what appears on the screen is the final result) because we want to optimize the time needed by the sketching step. Thus we need to create and visualize metaballs in real-time while at the same time this technique is very computationally expensive. We will show in the section 3.4 how we bypassed this issue.

In the last three paragraphs, we have extracted somehow, some important points to create a quality interface for this 3D organic shapes sketching system: it should be immersive (two hands), intuitive (force feedback) and using metaball primitives (to get what we see). Now, we will discuss another problem about design and sketching that is related to the animation of 3D characters. The both problems, i.e. 3D shape sketching and 3D animation sketching, are comparable as we will present it in the next paragraphs.

While animation designers can quickly and easily sketch 2D figures, the existing animation systems are not well suited for rapidly posing articulated 3D figures [18]. In this paper, we didn’t want to create a new animation technique, but to bring the immersive and intuitive effect of the interface described above to the animation of 3D characters. Thus, we choose a technique based on key frames and skeletons that can be found as a classic animation functionality of commercial 3D software as 3DSMax or Maya.

The next section of the paper will be dedicated to the system architecture which includes three hardware components: the Haptic Workstation™ for the interaction and the force feedback generation, a 3D visualization software using a Head Mounted Display (HMD) and the modeling tool based on metaballs.

### 3. SYSTEM DESCRIPTION

In this part, we will describe the hardware and the software that are used to build the haptic modeling and animation system. The figure 1 presents the main modules that are described along this section.

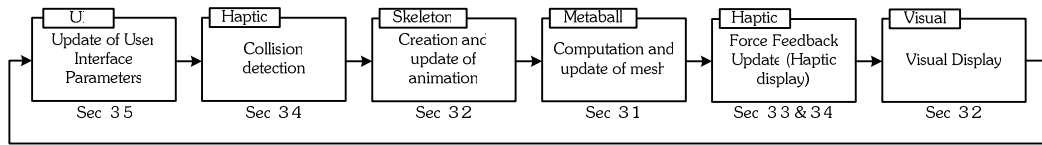


Fig. 1. Main loop of the animation and modeling system. Each module refers to section where it is described.

First, we introduce the metaballs and the skeleton animation modules. Then, we describe the VR hardware including the main interaction peripheral: the Haptic Workstation™. The following subsections describe the software modules which were used to implement the system: the haptic software and the user interface.

#### 3.1 Creation and Visualization of Metaballs

The metaballs are the main/only drawing primitive of the sketching system. This subsection describes the algorithm for calculating and visualizing them. We can split this explanation into two parts: the first one deals with the computation of the scalar field, and the second one describes the triangulation of the surface of the metaball.

The workspace is a scalar field where each point  $P$  of the space is characterized by a value  $x_p$ . In Figure 2, a high value of the field is represented by a dark color, and a low value is characterized by a light color. In our implementation, this value  $x_p$  is always positive.

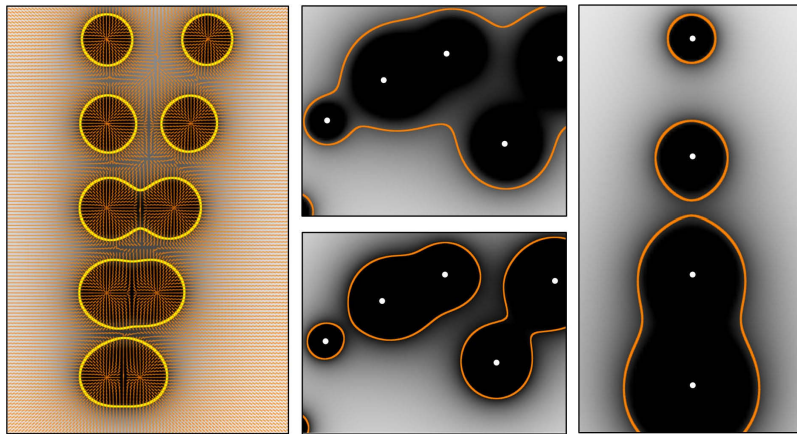


Fig. 2. Metaballs in a 2D space. The clear points represent the center of the metaballs. The scalar field is characterized by the gradient. The lines are the implicit curves, results of the algorithm; in 3D, these curves become surfaces. On the left, metaballs with same radius mixing together; on the center, effect of the influence parameter  $K$  on the resulting shape. On the right, effect of the radius parameter.

Each metaball  $M_i$  is characterized by a point, its center  $C_i$  (see figure 2) and by its influence on the field, a value  $T_i$ . For each metaball, the equation Eqn. (1) presents the function  $f_i(P)$  which gives its influence on the force field. The formula Eqn. (2) shows that for each point  $P$  of the field, we find its value  $x_p$  by adding the influences of each metaball.

$$f_i(P) = \frac{T_i}{1 + \text{distance}(P, C_i)^2} \quad (1)$$

$$x_p = \sum_{M_i} f_i(P) \quad (2)$$

To put in a nutshell, a metaball is a point that influences a field. By analogy, we can also imagine curves and even volumes which influence it. Our implementation contains also another primitive: the metaline, defined by two points, i.e. a segment. This primitive is very useful to create lengthened shapes, like the legs of a virtual character. Until here, we explained how to create a scalar field and modify it using primitives (points and lines, but this could be easily extended), now we will explain how this scalar field could be used to create surfaces.

We search for visualizing an implicit surface consisting of all points satisfying the function  $f(P)=K$ , where  $K$  is a parameter whose value could be choose by the user. This parameter influences the general shape of surface. A high value of  $K$  tends to bring the surface closer to the center the metaballs, while a low value tends to create a greater and smoother volume.

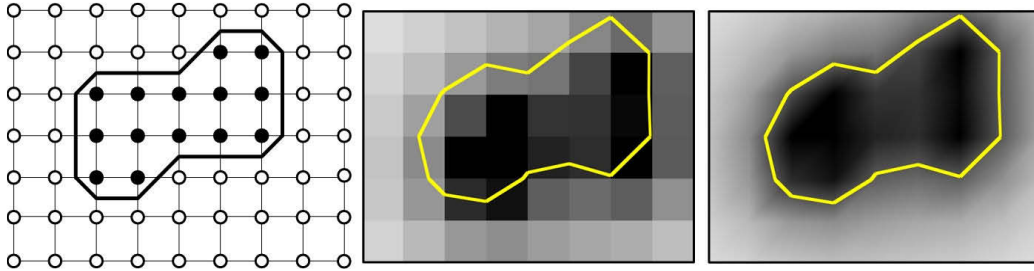


Fig. 3. Execution of the “Marching Cubes” with a discreet 2D field in  $8 \times 6$  squares. On left, the vertices of each square are evaluated using a binary knowledge (inside/outside of the metaball). On the center and on the right, the curve results from an interpolation between the true values of the field.

In this state, this surface cannot be rendered, since it is described by a continuous function. It must be triangulated. Thus we should have a discreet space, because it is impossible to calculate the value of the field at every points of the space, as we cannot find all the points  $P$  satisfying the function  $f(P) = K$ . The solution such as it was implemented in 1982 by Blinn, is still not real time. The algorithm of sampling used is the “Marching Cubes” [21] (see figure 3, for a 2D field). The principle is to divide the workspace into small cubes, then to analyze the values of the field at each vertex of these cubes (our implementation uses a discretization of  $64 \times 64 \times 64$  cubes). If a cube contains only vertices whose values are higher than  $K$ , the cube is inside a metaball, whereas if all values are lower than  $K$ , it is outside. Therefore, in these two cases, the cube does not intersect the implicit surface. In fact, only the cubes that have some vertices lower than  $K$  and some bigger than  $K$  have to be evaluated for creating the surface. Then there is two ways to build it as shown on figure 4 for a 2D field: either by having the binary knowledge that a vertex is inside or outside the metaball, or by taking in account the true value of the field and doing an interpolation. The first has the advantage of being a bit faster while the second, that we chose, better approximates the surface and is visually smoother.

### 3.2 Skeleton and Key-Frames Animation

In this subsection, we present the chosen system for animating the virtual characters. First, we define the notions of skeleton and bones. Then we present how we combine them with the metaballs using a technique that is inspired from mesh skinning.

In concrete terms, the bone of a skeleton is a line between two points. Each bone belongs to a hierarchical structure that represents the skeleton itself: this means that at least one extremity of a bone has to be linked to another bone, or to be the root extremity. This is a classic representation that is used in computer animation standards like H-AMIN (<http://www.h-anim.org/>). Then, this structure is usually used to deform a mesh by skinning. However in our case the skinning is achieved using only metaballs and metalines. Thus, we apply some restrictions to have a better control on the deformation: a bone extremity has to be set on the center of a metaball or on the extremity of a metaline. It means also that the bones in our system have fixed-length. To animate a bone, the user has to select one of its extremities, and to move it. The extremity of the bone must be always at the same distance from the other extremity, thus it can only be moved along the surface of a sphere.

Finally, we can animate this skeleton over the time using key frames. This implies to use a timeline widget in the interface. The relative positions/orientations of the bones are registered at each timestep of the timeline in order to store the animation. As mentioned in the state of the art, this technique is very similar with the one used in commercial modeling and animating software. The difference mainly comes from the user interaction with the software as it will be describe in the subsection 3.5.

### 3.3 Hardware Devices: Haptic Workstation and Head-Mounted Display

Our Haptic Workstation™ is a commercial product provided by Immersion™ [15] (see figure 4). This kind of haptic device is the only one that is able to create a complete force feedback on the wrists and on the hands. The user is then able to interact with the objects with a naturally more intuitive way.

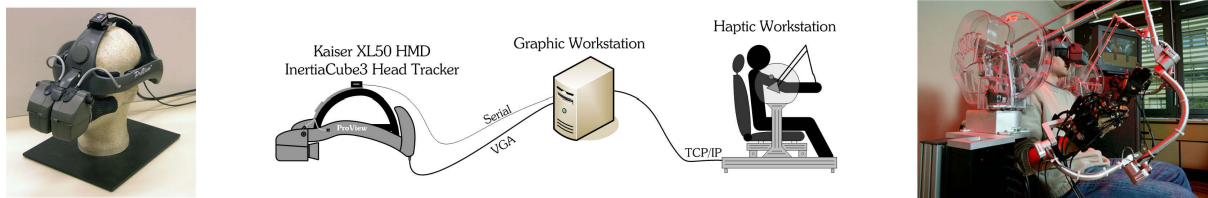


Fig. 4. Hardware architecture of the system.

In fact, the Haptic Workstation™ is simply bringing together several peripherals: the CyberGlove™, the CyberGrasp™, the CyberTrack™ and the CyberForce™. The pair of CyberGlove™ is used to acquire the hand posture. On each hand, twenty-two angles are evaluated, on all phalanges and finger abduct, making possible to have a good representation of the current hand state. The pair of CyberGrasp™, exoskeletons of the hand, is able to pull each finger separately using wires (however, this device cannot constrain the user to close one finger). They could be used to simulate the grasping of an object; the exoskeleton prevents from closing the fist by applying a force of 10N maximum. Two CyberTrack™, encapsulated into the CyberForces™ exoskeletons, allow gathering the position and the orientation of the wrist. The precision and the refresh rate of these values are much higher than using a magnetic or an optical system (for positioning, the accuracy is around the tenth of millimeter and around the tenth of degree for orientation; these values are updated at approximately 1000Hz). Finally, two CyberForce™ can be used to apply a three-dimensional force (maximum 80N) on the wrist, modifying the position of the user's hand if he does not resist. It is a 3DOF device because it can not be used to constrain the user to modify its hand's orientation.

Our sketching application does not use all these peripherals according to the standard usage. We will see indeed in part 3.5 that we made the choice to create an interface where all the elements are used: in other words, we use the peripherals only if they could bring information to the designer. For example, the manner for interacting with the interface does not require the representation of the real hand by a virtual hand faithful to reality (a palm, five fingers). The gloves are in fact used to detect particular postures corresponding to events in the interface, but not to model a hand. Thus, the force feedback on fingers, conveyed by the CyberGrasp™, is not used to simulate the grasping: we use it to give initiatives to the user, so that he/she could make a particular posture of fingers, according to the context (cf. §3.5).

Finally, this hardware system would not be complete without a visualization device. We chose to use a Kaiser ProView XL-50 (see figure 4), combined with an InterSense tracker which allows evaluating its orientation. In the following subsection, we deal with the software used for accessing and managing to these devices together.

### 3.4 Haptic Software

Our application must be able to take advantage of the Haptic Workstation™ and of the HMD for increasing the intuitiveness and the immersion of the designer. Usually, the Haptic Workstation™ is programmed using the VHT (Virtual Hand Toolkit) library [16] provided by Immersion™. This library includes many functions allowing to organize a virtual environment made of simple primitives (spheres, cylinders, cubes and convex volumes) and to manipulate them using the station. However, in our case, most of the objects have a dynamic shape, and it exceeds the capabilities of VHT. Thus, we chose to use our own library, MHAPTIC. This library allows a fast calibration, and a synchronized access to the data of the devices. Moreover, it contains useful modules like one to improve user comfort, or one for detecting collisions and computing force-feedback response. In this part, we present the calibration protocol, then the

module to improve comfort, and finally the manner to access to the data needed by the user interface module that is presented in section 3.4.

One of the most important points to avoid a break in the immersion when combining a haptic system and a HMD is to coordinate user movements with visualization. To achieve this goal, we have presented in [25] an algorithm to gather the data of the two CyberTrack™ with the same coordinate system. By default, the Haptic Workstation™ is not calibrated. This could be verified for example by joining the real hands: the visual result does not correspond to the reality, the hands are far the one from the other. In this organic shapes sketching system, we have re-used this module, and moreover, we have added to it the support of the head tracker. Thus, we have a good estimation of the hands and head position/orientation, in a single space.

Thus, the visual environment built is relatively intuitive for the user: the virtual camera moves according to the head movements, and the hand movements are correctly reproduced in the virtual environment. The visual immersion is then eased. However, in this situation, the user is still wearing an exoskeleton whose weight is not negligible, although it is compensated mechanically with counterweights. Moreover, this overload is not constant over the space: it is a function of the hand position. This implies that this device is uncomfortable to use during long sessions. The main reason is the posture of the arms, which must be kept outstretched horizontally while supporting the weight of the CyberForce™ (nearly 600g on each arm). Of course, this leads to a break in presence.

In [25], we have proposed a solution that consists in compensating for the weight of the exoskeleton (and also the weight of user arm if needed) using the CyberForce™ device itself. This method consists in using a precalculated force field. This force field contains, for each position of the hands, the vertical force value to apply for zeroing the exoskeleton weight. Thus, the virtual environment gains in realism: although it is not a visual improvement, the user does not need anymore to explain himself this overload. Of course, the feeling "to be imprisoned" in an exoskeleton does not disappear, but the tiredness does. We have shown, using bio-feedback measures, that this system considerably increases comfort, and, as consequence, immersion.

Finally, we have to deal with the collision detection system. It is mainly used to detect the collisions between the hands and the extremities of the bones, but it works also for moving the timeline widget (see section 3.2), and for rotating the 3D workspace. Each bone is linked with two spheres that are registered in the collision detection system. These spheres move when the hand touches them. This mechanism is also part of the MHAPTIC library. In the next part, we present the user interface that links the hardware and software components described in the four last subsections.

### 3.5 User Interface

This part presents the interaction paradigm which makes possible to create a 3D sketch, and to animate it. We saw how to parameterize the metaballs. Here is a summary list of goals that our interface has to achieve:

- Create a metaball.
- Create a metaline.
- Change the radius of a metaball (its influence on the field in fact).
- Change the radius of a metaline.
- Select a metaball center or a metaline extremity.
- Create a root node for skeletal animation.
- Create a bone.
- Move an extremity of a bone.

We chose to use posture of the hands to trigger the events, as we use the finger to click with a mouse. There is many advantages of using fingers: it could be done quickly, without moving a hand; and quickness is one of the main goal when dealing with sketching applications. There are four different postures: the hand closed, the index outstretched, both index and medium outstretched, and the hand opened. These postures are enough different to avoid confusions (even evaluating only the angles between phalanges). The state machine shown on figure 5 presents the protocol for triggering events cited above. The finger force feedback via the CyberGrasp™ device is used to help the user to know how he could reach the next state: for example, if he is in the "rest state" and that he opens the two fingers of the right

hand, the CyberGrasp™ will slightly drag the left hand fingers to help him to pass in the next state "Preview a metaline". The figure 6 presents a screenshot of the interface that the user views in the HMD.

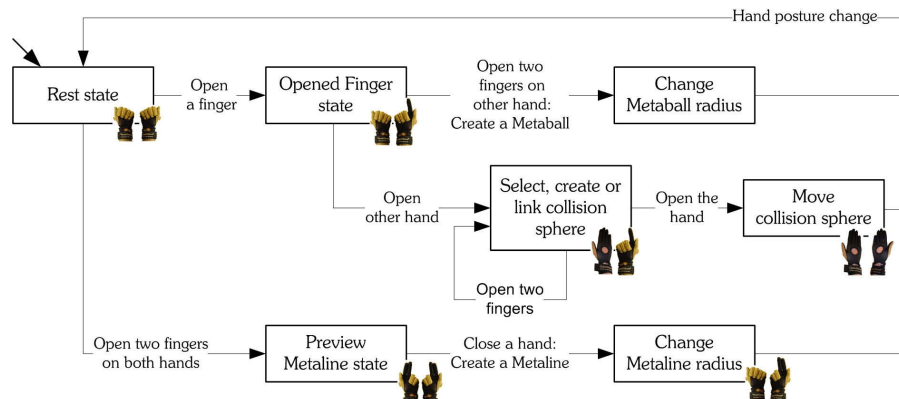


Fig. 5. State machine of the user interface.

The way to adjust the influence of a metaball or a metaline is very easy. Once that one is created (i.e. the center or extremities are defined) the radius of a metaball changes according to the distance between the finger and the center. The visualization remains real-time, thus the designer can choose the influence according to the results he wants to reach. However, after use, it seems to be difficult to choose the radius with precision, especially when the user has moved his hand far of the center and that he wants to do a small metaball, i.e. to put back his finger near the center. It is precisely in this kind of situations that the force feedback is useful. The idea is to define the center of the metaball as an attractor of the finger, and to change the force proportionally with the radius. Thus, if the user wants to create a metaball with a large radius, he must exert an important force, whereas he just needs to relax his effort in order to draw a small metaball. We have chosen a force of  $0.2N.cm^{-1}$  of radius. Since the workspace whose shape is cubic sizes 80cm, it represents a maximal force of 16N.

In this section, we have described the general architecture of the system, composed of VR devices, a metaball module used to create the mesh, a skeleton module for the animation and a specific user interface that takes advantage of the hardware. In the following section, we will present its evaluation.

#### 4. RESULTS AND EVALUATION OF THE TESTS

In this section, we present the evaluation of the metaball-based sketching system. In particular, we will see how the use of an immersive interface and haptic technologies contribute to the improvement of the system. Lastly, we will study the complete behavior when continuously used. Generally, the people having tested this interface are professional computer graphics experts, who have a perfect knowledge of specific software like 3D StudioMax or Maya. But, we also retained the opinion of computer scientists having only knowledge of the principles of modelling. We asked the designers to evaluate the interface in term of time to learn it, facility of use, and finally to evaluate whether we met the fixed goal: "to create a system allowing to carry out a draft of a geometrical model of human or animal shape which could be refined later using another software".

According to our study, the paradigm of interaction is not completely intuitive, particularly for the hand posture at the time of the creation of a metaline. Indeed, if it appears logical to have to tighten a finger (like to show something) to position the center of a metaball, then to close it to validate the creation of the metaball, it is not the same for the line: to tighten the index and the middle finger of both hands does not seem to have a real meaning. Moreover, the user get confused between the creation of metaballs and the selection of a collision shape. However, once the user knows well these postures, the flexibility in use represents a large advantage compared to the traditional interfaces. In particular, the majority of the questioned people found extremely good to have their model permanently with them, i.e. not to have to move their hands to manipulate components of interfaces (like buttons, sliders, etc). Moreover, concerning the immersive interface, using the HMD, the trackers on the hands and the force feedback, several comments were made. Initially, this mechanism is intuitive: the questioned people realize quickly that the real hands move the virtual hands, and that a movement of the head directs and moves the camera as in reality. Moreover, the force feedback in

particular to parameterize the radius of the metaballs offers a real help to the user. Indeed, the fact of attracting the hand towards a point according to its distance makes possible to very quickly give information of relative position. This remark is not surprising, P. Fuchs et al. already stated it in [12] while pointing out that the processing of visual data is more complex and longer than the processing of proprioceptive information.

Lastly, the system seen in a general way is interesting compared to the fixed goals. One can very quickly draw and visualize forms, but it should be admitted that a certain time of training (approximately ten minutes) is necessary to manage to model what one wishes. That rises directly owing to the fact that implicit surfaces function with control structures, like the splines. In fact, the place where the points of controls are placed does not correspond to the place where the surface passes: it is necessary to extrapolate the wished form, to find the good center. However, concerning the problem of the interactive modeling which was not inevitably ensured because of the inherent nature of implicit surfaces, the system remains real-time until approximately one hundred metaballs, which is large enough to carry out a draft of a geometrical model as shown in Figure 6. Ultimately, a computer graphics expert will probably prefer his/her usual software to carry out an unspecified model, even if he/she will admit that this system is faster under certain conditions (draft of Virtual Humans or organic objects). However, a relatively beginner who is not expert of any modeling software will prefer to carry out models with our system, because of its intuitive operational mode and its speed.

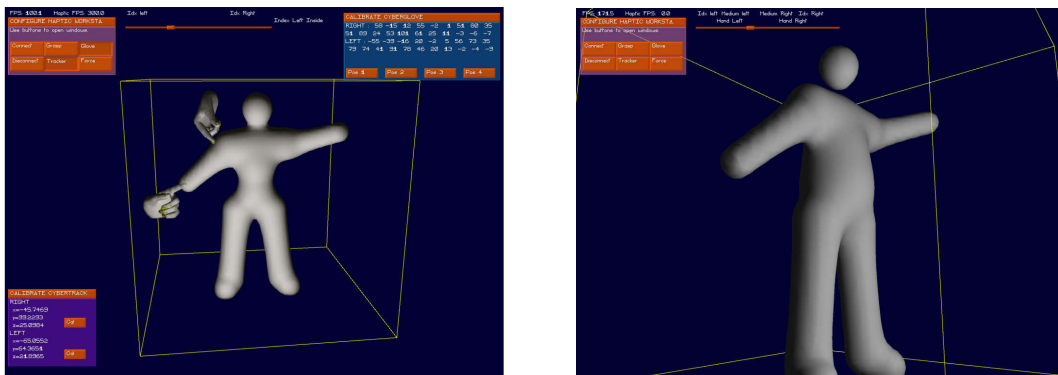


Fig. 6. Screenshots of the user interface through the HMD.

## 5. CONCLUSION AND FUTURE WORK

In this paper, we explained how to design and implement a geometrical modeling and animation system with an immersive interface using virtual reality devices. The principal assets of this interface are the use of the two hands, the force feedback based on the Haptic Workstation™, and the HMD, which thus make it possible to have a very intuitive paradigm of interaction. The duration of training, principal barrier with the use of the traditional systems of modeling, is some thus considerably reduced. This paper also shown that metaballs are an effective drawing primitive for carrying out drafts. Today, machines make indeed possible to have a real-time implementation of a visualization algorithm for implicit surfaces, which decreases the effect of their main drawback: global control of the shape. In a near future, we intend to integrate the devices used and the paradigm of interaction as a plug-in of a commercial modeling software that provides metaballs as primitives.

## 6. ACKNOWLEDGEMENTS

This work has been partially supported by the Swiss Federal Office for Education and Science under the framework of the European IST-Networks of Excellence AIM@SHAPE.

## 7. REFERENCES

- [1] Alexe A., Gaidrat V., Barthe L., Interactive modelling from sketches using spherical implicit functions, *Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa, AFRIGRAPH '04*, ACM Press, 2004, p. 25–34.
- [2] Alias®, Maya 6, 3D Modeling, Animation, Effects, and Rendering solutions, <http://www.alias.com/eng/products-services/maya/index.shtml>



- [3] Allen J. B., Wyvill B., Witten I. H., A Method for Direct Manipulation of Polygon Meshes, *Proceedings of Computer Graphics International '89*, 1989, p. 451–469.
- [4] Angelidis A., Cani M.-P., *Surfaces implicites multi-résolutions*, Master's thesis, Institut National Polytechnique de Grenoble, France, 2001.
- [5] Bernstein N. L., Lawrence D. A., Pao L. Y., Design of a uniaxuated bimanual haptic interface, *Proceedings of Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2003, p. 310–317.
- [6] Blinn J. F., A Generalization of Algebraic Surface Drawing, *ACM Transaction on Graphics*, ACM Press, Vol. 1, No 3, 1982, p. 235–256.
- [7] Bloomenthal J., Wyvill B., Interactive techniques for implicit modeling, *SI3D '90: Proceedings of the 1990 symposium on Interactive 3D graphics*, ACM Press, 1990, p. 109–116.
- [8] Calinon S., Vexo F., WiCKed : Sculpture on Virtual Head, *student project EPFL*, 2002.
- [9] Deinsinger J., Blach R., Wesche G., Breining R., Simon A., Towards Immersive Modeling - Challenges and Recommendations : A Workshop Analyzing the Needs of Designers, *Proceedings of ISATA '00*, Dublin, Ireland, 2000.
- [10] Discreet®, 3D Studio Max, Animation, Modeling and Rendering software,  
<http://www.discreet.com/products/3dsmax/>
- [11] Foskey M., Otaduy M. A., Lin M. C., ArtNova : touch-enabled 3D model design, *Proceedings of the IEEE Virtual Reality Conference*, 2002, p. 119–126.
- [12] Fuchs P., Moreau G., *Le Traité de la Réalité Virtuelle*, second release, vol. 1, *Les Presses de l'Ecole des Mines de Paris*, 2004.
- [13] Galyean T. A., Hughes J. F., Sculpting: an interactive volumetric modeling technique, *Proceedings of the 18th annual conference on Computer graphics and interactive techniques SIGGRAPH '91*, ACM Press, 1991, p. 267–274.
- [14] Hsu W. M., Hughes J. F., Kaufman H., Direct manipulation of free-form deformations, *Proceedings of the 19th annual conference on Computer graphics and interactive techniques SIGGRAPH '92*, ACM Press, 1992, p. 177–184.
- [15] Immersion Corporation, Haptic Workstation™ description,  
[http://www.immersion.com/3d/products/haptic\\_workstation.php](http://www.immersion.com/3d/products/haptic_workstation.php)
- [16] Immersion Corporation, Virtual Hand SDK datasheet,  
[http://www.immersion.com/3d/products/virtualhand\\_sdk.php](http://www.immersion.com/3d/products/virtualhand_sdk.php)
- [17] Kuo-Cheng W., Fernando T., Tawfik H., FreeSculptor : a computer-aided freeform design environment, *Proceedings of the 2003 International Conference on Geometric Modeling and Graphics*, 2003, p. 188–194.
- [18] Lasseter J. , Tircks to Animating Characters with a Computer, in *SIGGRAPH 94 Course Notes No. 1*. Orlando Florida, 1994.
- [19] Lawson B., *How Designers Think: The Design Process Demystified*, 3rd ed., *Butterworth-Heinemann*, Oxford, 1997.
- [20] Lipson H., Shpiltani M., *Product Development and CAD/CAM*, Verlag, Berlin, F.L. Krause, 1998.
- [21] Lorensen W. E., Cline H. E., Marching cubes: A high resolution 3D surface construction algorithm, : *Proceedings of the 14th annual conference on Computer graphics and interactive techniques SIGGRAPH'87*, ACM Press, 1987, p. 163–169.
- [22] Magnenat-Thalmann N., Panfzic I. S., Moussaly J.-C., The Making of the Terra-Cotta Xian Soldiers, *Proceedings of Digital'97, Digital Creativity*, Geneva, 1997, p. 66–73.
- [23] Magnenat-Thalmann N., Thalmann D., *Handbook of Virtual Humans*, John Wiley and Sons, 2004.
- [24] McDonnell K. T., Qin H., Virtual Clay: Haptics-based Deformable Solids of Arbitrary Topology, *Proceedings of the Second International Workshop on Articulated Motion Deformable Objects*, 2002.
- [25] Ott R., Gutierrez M., Thalmann D., Vexo F., Improving User Comfort in Haptic Virtual Environments trough Gravity Compensatio, *Proceedings of the First Joint Eurohaptics Conference And Symposium On Haptic Interfaces For Virtual Environment And Teleoperator Systems*, Pisa, Itilia, 2005.
- [26] Perry R. N., Frisken S. F., Kizamu : a system for sculpting digital character, *Proceedings of the 28th annual conference on Computer graphics and interactive techniques, SIGGRAPH '01*, ACM Press, 2001, p. 47–56.
- [27] Salisbury J. K., Srinivasan M. A., Phantom-Based Haptic Interaction with Virtual Objects, *IEEE Computer Graphic Applications*, IEEE Computer Society Press, vol. 17, no 5, 1997, p. 6–10.
- [28] SensAble®, SensAble Technologies Inc., FreeForm™ modeling system,  
<http://www.sensable.com/products/3ddesign/freeform/hardware.asp>

- [29] Shen J., Thalmann D., Interactive Shape Design Using Metaballs and Splines, *Proceedings of Implicit Surfaces '95*, Grenoble, France, 1995, p. 187–196.
- [30] Thalmann D., Simulating a Human Society: the Challenges, *Proceedings of Computer Graphics International, CGI'02*, 2002.
- [31] Thalmann D., Vexo F., MPEG-4 Character Animation, *KI*, vol. 17, no 4, 2003.
- [32] Tovey M., Porter S., Newman R., Investigating automotive sketches as a source of geometry for 3D models, *Proceedings of the Tools and Methods of Competitive Engineering Symposium*, China, 2002.