The over-sketching technique for free-hand shape modelling in Virtual Reality

Fabio Bruno, Maria Laura Luchi, Maurizio Muzzupappa, Sergio Rizzuti

Università della Calabria
Facoltà di Ingegneria, Dipartimento di Meccanica
Ponte Pietro Bucci 44/C, 87030 Rende (CS) Italia
Tel.: +39 0984 494601 – Fax: +39 0984 494673
e-mail: {brunof, luchi, muzzupappa, rizzuti}@unical.it

Abstract : The benefits of the Virtual Reality (VR) technologies in the field of conceptual free form modelling have been investigated for over a decade. Many researchers have paid attention to hand-sketching techniques for shape generation and to the Virtual Sculpting approach for surface modification. What has not yet been investigated sufficiently is modification based on the free-hand drawing approach. This could be more similar to the traditional way designers work, compared to the sculpting metaphor.

The research described in this paper helps to fill this gap, defining a new free-form curve modification technique based on the over-sketching approach. This technique has been implemented in a curve modification tool of VEIS (Virtual Environment for Interactive Sketching), a prototype VR system for free form modelling. The tool allows the user to refine a curve by simply substituting a part of it with a new one. It recognizes the user gestures, made with a 3D tracked pen, to determine how the pen strokes modify a part of the existing curve and then automatically merges the new piece with the unchanged parts of curve. Furthermore, by taking advantage of the parent-child relation, a surface can be updated by simply modifying the curve/s employed for its definition.

Key words: Free-hand sketching, free-form surface modelling, 3D input device, human-computer interaction, virtual reality.

1- Introduction

The introduction of Virtual Reality (VR) into the conceptual phase of the design process gives the designer the possibility of creating and manipulating 3D shapes as if these were really in front of him. There are already some systems that aim at the exploration of this important research field. In our opinion, the most promising application of modelling in Virtual Environment is the sketching and the modification of conceptual free-form shapes. In this area, in fact, the VR can make a useful contribution to the work of the designer making the conceptual modelling more intuitive and interactive.

In the first part of the paper we give a brief description of the VEIS system and of its functionalities for curve and surface sketching (for more details see references: [1], [2], [3]). In particular we describe the hardware configuration, the software architecture of the system and the interaction techniques employed for the sketching tools. In the second part we pay attention to the modification techniques. This represents the major contribution of the paper because the “simultaneous correction” and the “over-sketching” concepts for curve modification have been introduced and combined in a VR modelling system. Taking advantage of the parent-child relation, a surface can be updated by simply modifying the curve/s employed for its definition.

In section 5 we describe the over-sketching technique and its implementation in VEIS in detail. Over-sketching is more intuitive and natural than the traditional curve modification methods (based on the manipulation of the control points or the interpolating points), because it is more similar to the hand-drawing technique. Moreover it could be faster because the user only has to re-sketch the part of the curve that he/she wants to change. VEIS, like many VR modelling systems, is specifically aimed at the quick and intuitive modelling of conceptual shapes. Combining over-sketching with “simultaneous correction” and with the revisited interaction techniques for surface sketching, we try to demonstrate that the modelling of conceptual free form shapes can be significantly improved by the VR technologies.

2- Previous Work

Some interesting research has already been undertaken which aims to define tools and methodologies for shape modelling in Virtual Environment.

The first approaches, such as 3-Draw [4] and JDCAD [5], were based on normal desktop configurations equipped with a pair of shutter glasses for stereoscopic vision and a magnetic tracking device for interacting with the system. JDCAD supports creating 3D primitives and performing Boolean operations. In 3-Draw the user sketches a set of curves and uses them to define the skeleton of a surface, as in

a wireframe representation, but there is no possibility of creating a surface with the curves drawn. One of the first modelling applications using immersive Virtual Reality technology was 3DM [6]. It uses head mounted displays to visualize surfaces modelled by triangle nets.

COVIRDS [7], Conceptual Virtual Design System, tries to combine various VR-technologies, such as speech input, gesture recognition as 3D input and output during the conceptual phase and to develop new interaction techniques for creating and modifying free-form surfaces. Modifications can be performed with the so called virtual tools: with a cone tool, for example, it is possible to create a hole in a surface.

In ARCADE [8] and in SpaceDesign [9] the 3D free-form surface modelling is implemented in different ways, e.g. Coons patches from just one 3D outline stroke, skinned surfaces with immediate visual feedback during creation, symmetric free-form surfaces by using the pad as a mirror plane, subtractive sweeping, etc.. But there is no possibility of changing the curves previously drawn and only the trimming tool is available to modify the surfaces.

In the Surface Drawing system [10] the hand and a set of tangible tools are used to edit and manipulate 3D models. The system is particularly addressed at the creation of artistic shapes.

Liverani and Piraccini [11] developed a surface modification tool for the IRR (Immersive Reconfigurable Room), a CAVE-like VR immersive system. Moving a tracked glove over the surface, the user can "spread" a profile on the surface, and so model the surface, according to the glove movements and the tool shape.

Returning to the traditional CAD environment, Van Dijk et al. [12] have developed an algorithm for the input and modification of hand sketched profile curves. The interaction technique employed for the curve modification is based on the concept of over-sketching: the user can sketch an additional stroke that is digitised and merged with the old stroke. This algorithm has been incorporated in a prototype surface modeller.

Pernot et al. [13] have presented a modification technique for NURBS surfaces based on the manipulation of the character lines using a set of curves to define the control parameters geometrically constraining the initial surface.

A great deal of research has been devoted to developing a surface modification technique, based on the concept of Virtual Sculpting, suitable for 3D input devices. Virtual Sculpting allows shapes to be modified as if these were made of clay, and the user can change them using various tools. Ferley et al. [14], [15] have proposed a Virtual Sculpting system that can work either with a 3D input device or with the mouse. The user can employ a set of tools that allows the shapes to be added, removed and altered. This approach is based on implicit surfaces defined as iso-surfaces over a discrete field.

Perles and Vance [16] have developed a set of virtual tools which are used to operate directly on the CAD data and change the shape of NURBS surfaces in a virtual environment. The virtual tools can be dragged along or pushed into the surface in real-time to change its shape. The normal at an arbitrary surface point can be manipulated in three-dimensional space.

The Virtual Sculpting approach is very promising, especially in conjunction with the development of the haptic technologies, but it is important to underline that Virtual Sculpting is based on an interaction technique that reflects the typical gestures of sculptors. The Virtual Sculpting systems are intended to be used by people that are able to model clay who can be further stimulated by the potential of these interesting tools. However, CAD users could encounter problems with the sculpting metaphor, which requires a specific skill. Indeed, at the moment it is very difficult to control the accuracy of the modelling, and, furthermore, the development of techniques that allow a precise control of the shapes seems to be an unattainable goal.

3- Overview of the VEIS system

The approach followed for the development of the modelling tools in VEIS is based on the extension of concepts already known in the CAD systems, where the interaction technique has been revised to support an intuitive and interactive modelling, using 3D input devices. Our aim is to develop a system that helps us to demonstrate the possibility of improving the modelling technique currently employed in CAD systems, taking advantage of the Virtual Environment.

3.1– Hardware configuration

The hardware system on which VEIS has been implemented and tested belongs to the so called desktop VR configuration. Because current research is mainly concerned with the interaction technique for curve and surface modelling in Virtual Reality, it has not been essential to acquire an advanced Virtual Reality system, due both to the great expense and complex set-up. Usually in a Virtual Reality modelling system the user sketches the shapes using a 6 DOF tracked pen that can be used with all the modelling tools. In VEIS the virtual pen is used for interacting with the system can be controlled in three different ways: the Microscribe-3DX; the Spacemouse; or a mouse in conjunction with a construction plane which can be moved and rotated by the Spacemouse or by the keyboard. The Microscribe-3DX is a device based on an electromechanical arm able to obtain the position and the orientation of a pen placed at the end of the arm. It is commonly used in reverse engineering applications to build a 3D model from a real object. Compared to other 3D input devices, such as electromagnetic or optical tracker ones, the Microscribe-3DX has some limitations in its movement (the rotation around the pen axis is not allowed) and partially covers the monitor surface, but, on the other hand, it is very precise (0,18 mm), allows a quick set-up and is not very expensive. The user can also sketch the shapes with the Spacemouse, but it works like a joystick, and therefore is inadequate for the modelling phase; on the other hand it is really useful for rotating and translating either objects or the whole scene. The third modelling method is based on a construction plane that allows the application to be run on machines without 3D input devices, which is really useful for the developers. The software can work in full-

screen mode or in window mode, and the user can choose from several tools, using the mouse, through pop-up menus or through the buttons on the toolbar.

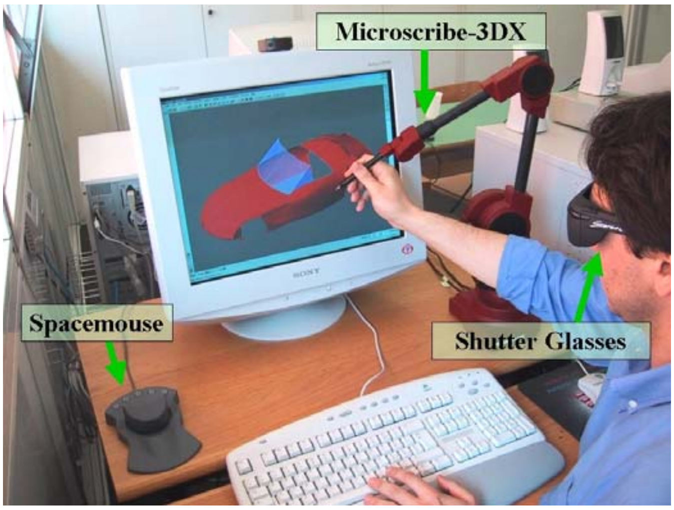


Fig. 1: The hardware configuration of VEIS

3.2– Software implementation

VEIS has been developed using two Software Development Kits: Open Inventor 3.1 by TGS for managing the 3D scene and for navigating in it, and Open Cascade, an open source modelling kernel. Open Cascade has been used to resolve some typical calculation problems related to the NURBS mathematics.

As described in Figure 2 VEIS is based on three C++ classes. VEIS_Shape is the abstract base class for the other classes. It is useful for managing the scene data base. It is also used for implementing some operations such as selecting, moving, copy & paste, etc. VEIS_Curve and VEIS_Surface include a dual representation of the curve and the surface data in Open Inventor and in Open Cascade. Some modelling operations are made using the Open Cascade shapes representation. In this case the Open Inventor data has to be continuously updated in order to show the changes of the shape. Other operations like translation, rotation, etc. are made directly on the Open Inventor data in order to have a better visualization performance. When the operation is finished the Open Cascade data is updated to maintain the consistency between the two representations.

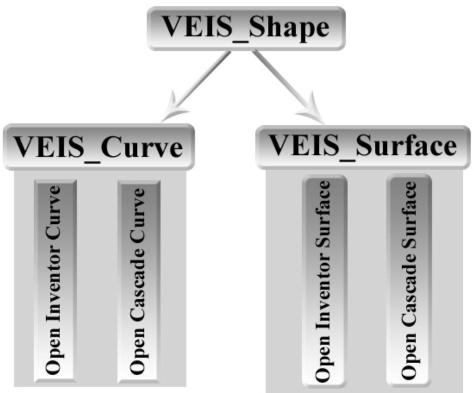


Fig. 2 : The software structure of VEIS

3.3– Curve sketching

The basic tool of VEIS is curve drawing, that allows the user to sketch a curve by simply moving the pen. This approach, that has been discussed in [17], is very different from those employed in the traditional CAD systems. In these, the user has to specify some mathematical parameters such as order, continuity, etc. and has to input a sequence of control points or a set of interpolating points. These systems require that the user knows the mathematical model of the curve (Spline, B-Spline, NURBS, etc.). On the other hand, in some cases like in conceptual design, the user does not have to control the curve mathematics, but it is very important for him/her to sketch and correct the curves quickly.

The algorithm employed in VEIS for curve sketching has already been described in [18]. It generates a non periodic B-Spline of order four, approximating the path defined by the pen motion. The algorithm captures a sequence of points, sampled by the tracking system, in order to composes the control polygon of the curve. The points are sampled at fixed time steps, so the control polygon can be less or more coarse depending on the speed of the pen movements. Furthermore the "simultaneous correction" of the curve has been implemented in VEIS following the 3D-Eraser Pen technique [18]. The key characteristic of the technique consists in the possibility for the user to correct a curve during the sketching, without interrupting the drawing process, modifying what has just been drawn by simply reversing the direction of the hand movement and without changing tool. The correction is done in real time and, if the pen inverts its direction again, the drawing process restarts. In this manner it is very simple to correct an error or modify the curve shape.

3.4– Surface sketching

The VEIS surface modelling tools are based on the extension of concepts already known by CAD users (extrude, skin and revolve). In order to make the sketching phase more interactive the surfaces are shown as soon as the user starts to sketch the shapes and these are updated at every movement of the pen.

3.4.1– The Extrude Tool

This tool allows the user to sketch a surface using the well known concept of extrusion. The user can draw a curve or select one of the curves already drawn. Then the curve is attached to the pen and the user can drag it. As soon as the pen is moved, the surface generation starts and its shape is immediately shown. During the extrusion, the 3D Eraser Pen can be used to correct the extrusion path, and as a result the surface is also corrected.

3.4.2– The Skin Tool

This tool generates a surface interpolating a set of curves that are drawn progressively. This technique is also known in literature as lofting. In our implementation the surface appears as soon as the user starts to trace the second curve and it is updated in real time whenever the pen is moved, even if the user employs the 3D Eraser Pen to correct the sketching. The speed of the interpolation algorithm depends

on the number of profiles and on their complexity. Obviously the surface updating speed decreases as new profiles are added. In any case, on a standard PC, it is possible to draw up to 6-8 profiles without the interactivity of the system decaying.

3.4.3– The Revolve Tool

This tool allows the user to generate surfaces of revolution. The user draws a curve or selects one from the curves already drawn, and defines the axis of revolution picking two points in space. Then the user, moving the pen around the axis, drags the profile on a circular path and observes the surface, which is progressively generated in proportion with the hand gesture. Also in this tool the surface is shown as soon as the designer starts to drag the profile and it is continuously updated until the end of the modelling session. The user can interrupt the revolving at any moment, can use an automatic snap in order to generate the surface at steps of 90° and, reversing the hand gesture, can delete a portion of the surface. Furthermore the user can delete the surface completely and redraw it starting in the opposite direction.

3.5– Support functionalities

Some support functionalities have been implemented in order to make the system more flexible. Using Open Inventor environment functionalities, the user can change the material of the shapes, manage the lights on the scene, change the point of view, etc.. Moreover we have been implemented a mirror plane that can be moved in the scene with the pen, the snapping of the pointer on the curve already drawn and the selection, translation and rotation of the shapes with the pen.

4– Curve and surface modification

The study of new interaction techniques for surface modification in Virtual Reality has two goals. On the one hand it is important to define a suitable interaction technique that could be used to implement a surface modification tool inside the next generation of Virtual Reality modelling systems. In fact in the systems previously described in section 2 it is impossible to modify the sketched surfaces by editing the construction curves. On the other hand it should be noted that Virtual Environments are currently employed for the design review of styling models created in a traditional CAD environment. However the refinements of the models have to be carried out in the CAD system, so it is not possible to see the effects of the changes immediately and directly in the Virtual Environment. The development of a suitable modification tool in the design review process would be very useful and could be done in a short space of time.

The surface modification functionality implemented in VEIS is based on the parent-child relationships between the surfaces and the generating curves. This approach, which is considered a powerful feature, has been implemented in the most advanced modelling software, but it has not yet been tested in a VR modelling system. Therefore an attempt to try this idea in a Virtual Environment has been done in VEIS. In fact in VEIS the surfaces can be modified by editing the parent curves: the four outline curves in the Extrude; the profile curve in the Revolve and each one of the lofting curves in the Skin. For the Revolve the curve describing the arc of revolution, and consequently the revolution angle, can also be changed. For the Skin surfaces it is also possible to translate and rotate a lofting curve by simply selecting a curve and moving it with the pen. Also in this case, the surface is updated at every movement of the curve. These kind of modifications do not change the typology of the surfaces e.g. if the user modifies the profile of a revolve surface, this continues to be a revolve surface. In the case where a curve or a surface is generated by the mirror plane, every modification made to it is reported directly on the symmetrical part, in order to maintain correspondence between the mirror objects.

For curve modification two different techniques have been developed. The first one uses the 3D Eraser Pen concept, described in the next section. The second technique is based on the over-sketching method.

4.1– Curve modification with 3D Eraser Pen

The first implementation of the modification tool is based on the 3D Eraser Pen. It allows the sketching process to be restarted from the two curve ends. In other words, the user can extend or shorten the curve dragging each one of the two tips and using the 3D Eraser Pen to try different sketches. The tool works as follows: the user selects a curve in the scene, chooses the modification tool from the toolbar and two spheres appear on the two ends of the selected curve. The user can now select a sphere with the pen and, moving it, restart the curve sketching, using the 3D Eraser Pen to make the corrections. Furthermore if the curve is related (as parent) to a surface, the surface will be modified in real time to follow the curve changes.

In figures 3, 4 and 5 a working sequence of the modification tool is shown. The surface in Figure 3 was created revolving the red profile of 270°. The user selects the profile and, as soon as he/she activates the modification tool, two spheres appear on the profile ends. Figure 4 shows how the user, moving the upper sphere back along the curve, deletes part of the curve and, as a result, part of the surface is deleted too. As shown in figure 5, as soon as the user reverses the direction of the pen movement, he/she restarts the sketch of the curve and the surface is immediately updated.

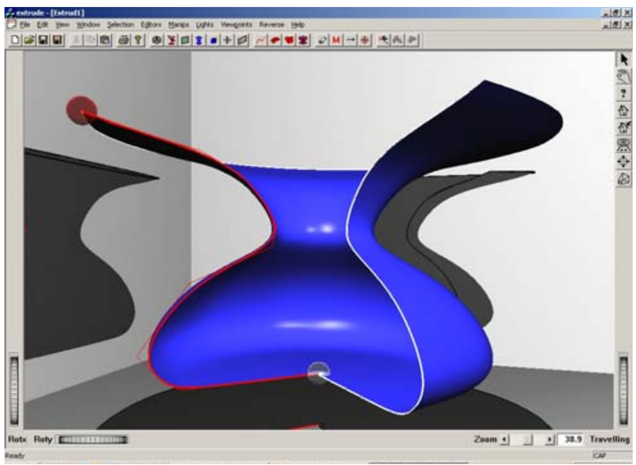


Fig. 3: The profile curve has been selected and two spheres appear on the two ends of the selected curve

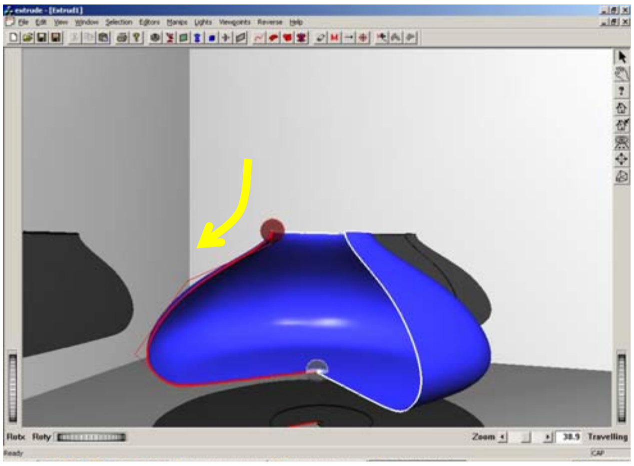


Fig. 4: The profile curve has been partially deleted, and the surface has been consequently updated

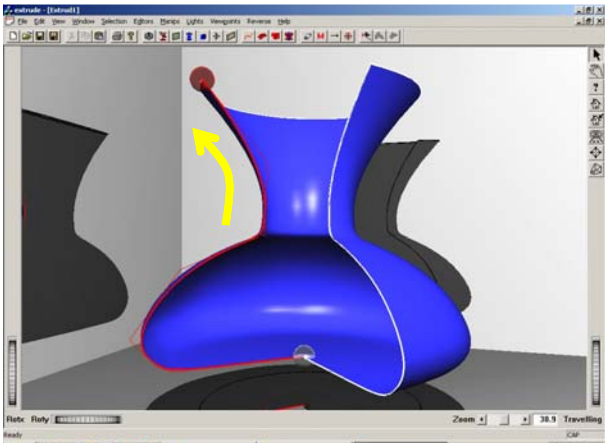
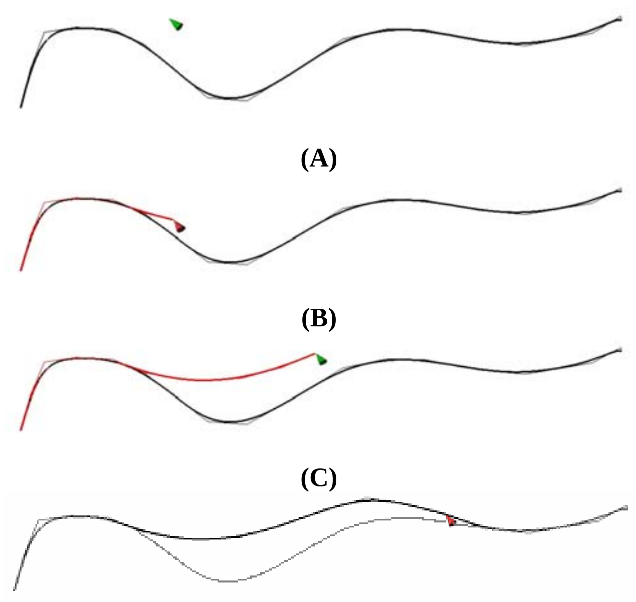


Fig. 5: The profile drawing restarts

5– Curve modification with over-sketching

Over-sketching is a technique for curve modification based on the possibility of refining a curve by simply substituting a part of it with a new one. In our implementation we have used this approach either for the middle or the ends of the curve present in the scene, previously drawn or imported from a file. Supposing that the user wants to modify a curve in the middle, he/she begins by selecting a curve. Then he/she activates the tool and moves the pointer near the curve (Figure 6-A). The pointer changes colour when it is near the curve, to indicate to the user that he/she can start to draw a new stroke from the current position. When the user presses the spacebar on the keyboard the new curve appears (the red one in Figure 6-B). This is built duplicating the left part of the old curve. Only a short part of the new curve is different from the old one because it has to reach the pointer position. This in fact can be located inside a small region (controlled by a threshold value) around the old curve. The joining strategy will be explained in section 5.2. Now the user can move the pen to sketch the new curve as described in section 3.3 (Figure 6-C). If the pointer is again moved near the old curve it changes colour indicating that, if the user presses the spacebar, the new curve will be joined to the old one in the current pointer position (Figure 6-D).



(A)

(B)

(C)

(D)

Fig 6: Four snapshots of a working sequence with the over-sketching tool

We have experienced that if the old curve is immediately deleted the user can experience difficulties in evaluating the results of the modification. This is probably due to the method commonly used in hand drawing. When a designer makes a correction it is probable that, at first, he/she draws a new stroke and then deletes the undesired part of the existing sketch. In our over-sketching tool we have tried to replicate this approach: as soon as the user finishes drawing the new curve, the old curve is not deleted but it is softened (Figure 6-D). In this way the two curves have a different colour and the user can evaluate the difference between the old curve and the new one. If the user makes a sequence of corrections on the same curve the colours of the old curves are gradually softened (Figure 7). When the tool is deactivated all the old curves are deleted from the scene. The maximum number of old curves shown by the system for a modification sequence has been set, as default, to 5. In any case the user can change this value or simply disable the visualization of the old curves.

Fig.7: The old curves are gradually softened

The system stores all the modifications that the user makes on the curves. So the user, using two buttons on the toolbar, can perform the classical undo/redo operations. Applying these common tools to our implementation of the over-sketching modification technique we have experienced that it is very useful for the designer to try different sketches of a curve, while continuing to be able to see the previous sketches displayed in graduated softened colours.

When the designer employs the over-sketching tool he/she has to take into account the direction followed in the old curve sketching. The user has to draw the new curve moving the pen in the same direction employed for the old one. This may seem to be a limitation, but, as described in [12], the merging strategy used to incorporate the new curve in the old one is a relevant problem in the over-sketching approach. Reference 12 describes how a modification can have up to 24 different results depending on the merging strategies adopted. In our opinion this ambiguity can be resolved by "forcing" the user to draw the new curve following the same parametric direction as the old one. In this the designer is helped by a set of arrows which appear as soon as the curve is selected (see the first snapshot of Figure 9).

The over-sketching tool can be used to modify the initial or the final part of a curve. In order to correct the initial part of the curve (as described in Figure 8-A) the pointer must be positioned sufficiently: a) far from the old curve when the user starts the new curve, and b) near the old curve when the user decides to finish the modification. Otherwise, to modify the final part of a curve (see in Figure 8-B) this sequence is reversed. In Figure 8-C the complete substitution of a curve in the scene is shown. This can be extremely useful when the designer has to modify a surface patch replacing a parent curve with a new one.

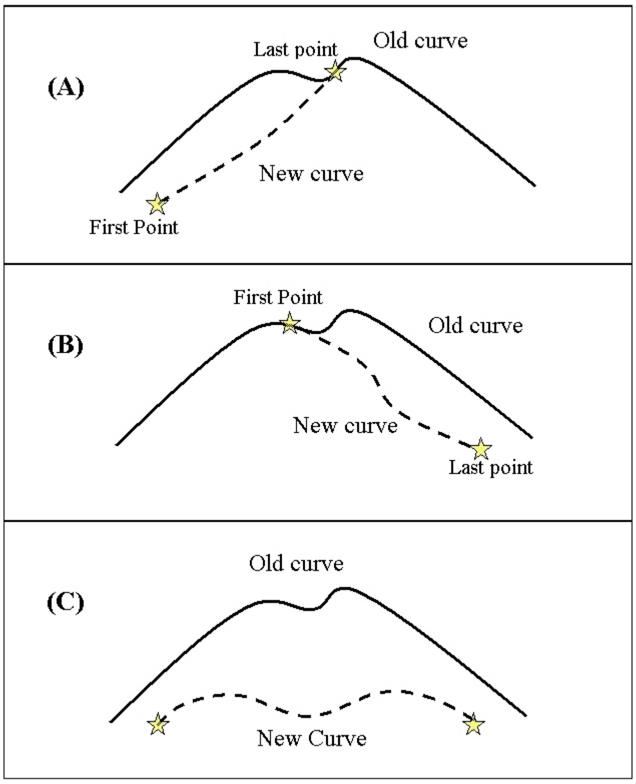


Fig 8: Merging strategies

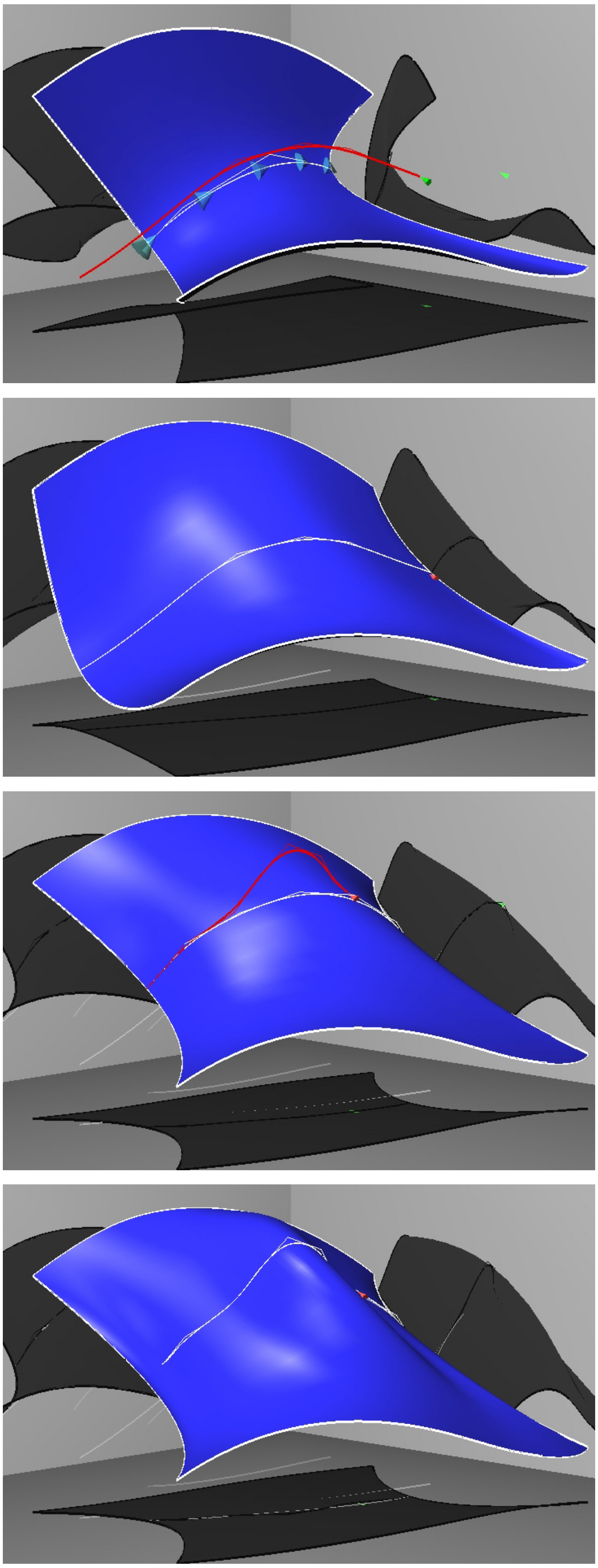


Fig. 9: A skin surface has been modified with the over-sketching tool. In the first snapshot the new curve substitutes the old one completely

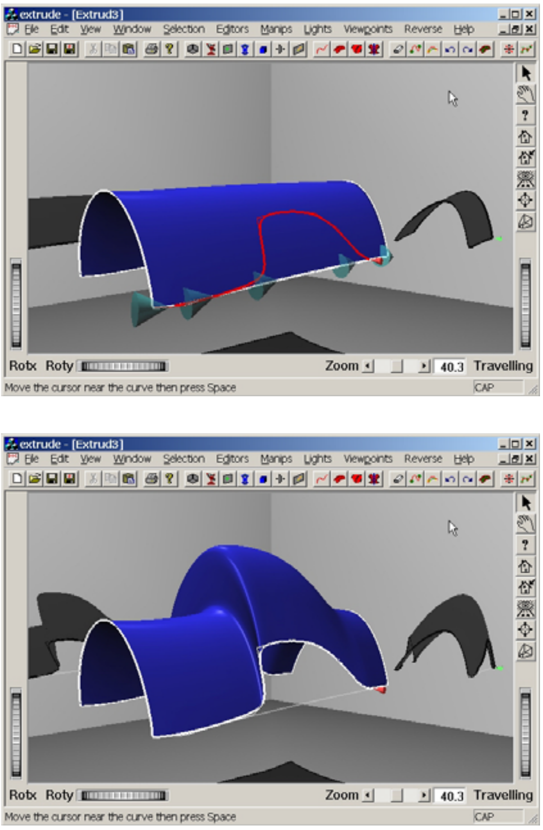


Fig. 10: An extrude surface has been modified with the over-sketching tool

5.1- Implementation of the over-sketching tool

The implementation of the over-sketching tool is based on two event callback (CB) functions: *press_event_CB* and *motion_event_CB*. The first manages the press button event and the other manages the pen motion event. A *state* variable has been introduced in order to control the system state, which assumes the *modifying* value if the user is drawing the new curve and, otherwise, it assumes the *stop* value.

The Open Cascade function called *ProjectPointOnCurve* has been used to calculate the point on the curve nearest to the pen position and the distance between them (*proj_distance*). If the distance is lower than a fixed threshold (that can be changed by the user) the new curve is joined to the old one. Another Open Cascade function (*Geom_BSplineCurve::LocateU*) calculates the knots u_1 *and* u_2 that define the parametric range containing the point on the curve nearest to the pen position. An implementation of these two functions can be found in *The NURBS Book [19]*.

Calling the value of the old curve parameter u corresponding to the point where the new curve starts u_i, the parametric range is (u_{i1}, u_{i2}); calling the value of u where the new curve ends u_f, the parametric range is (u_{f1}, u_{f2}). Figure 11 shows the initial and final parametric values u_i and u_f, and their parametric range, in the case when a curve is modified in the middle. The parts of the old curve defined by the parametric ranges [0, u_{i1}] and [u_{f2}, 1] are employed for the construction of the new curve. When the user starts the modification the u_i value and the lower bound u_{i1} are calculated. The list of control points [P(0), .. , P(j)] acting on the parametric range [0, u_{i1}] is copied on the control point array of the new curve. The user can now draw the new piece of curve and the motion events are managed by the *motion_event_CB* method of the *VEIS_curve* class. New control points are added to the new curve array. When the user decides to finish the modification, and approaches the old curve again, the u_f value and the upper bound u_{f2} are calculated. The list of control points [P(k), .. , P(n-1)] acting on the parametric range [u_{f2}, 1] is appended to the control point array of the new curve.

The condition about respecting the parametric direction can be expressed as: $\boldsymbol{u_i < u_f}$. The part of the old curve included in the parametric range (u_{i1}, u_{f2}) is substituted by the new curve. This constraint can also be extended to the cases when the initial or the final part of the curve have to be modified. In these cases we assume, respectively, $u_i = 0$ or $u_f = 1$.

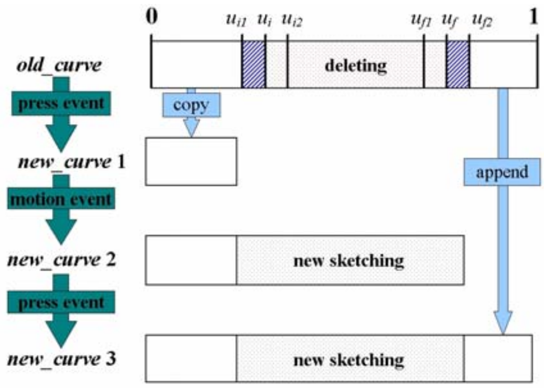


Fig. 11: *Diagram of the parametric ranges on the old curve and the new curve*

5.2- Joining the curves

In developing of an adequate strategy for joining the old curve and the new one we must trade off between two conflicting requirements:

1) Preserve the smoothness of the new curve.
2) Minimise the zones in which the new curve deviates from the old one, in proximity of the joints.

In figure 12 the snapshots show the moment in which the user has pressed the spacebar in order to start/finish the modification. As illustrated in figure 12-A, when the new curve is created, the final part of the new curve (linked to the pointer) deviates from the old curve. The blue zone shows this difference. Its length is related to the size of the parametric range [u_{i1}, u_i] that has been neglected for the construction of the new curve. The width of the blue zone is related to the distance between the pointer and the old curve. The same situation occurs when the new curve is terminated (see figure 12-B). In this case the region is related to the size of the parametric range [u_{f2}, u_f]. In order to reduce the length of the blue zones, a reparametrization of the old curve in the parametric ranges [0, u_i] and [u_f, 1] should be made.

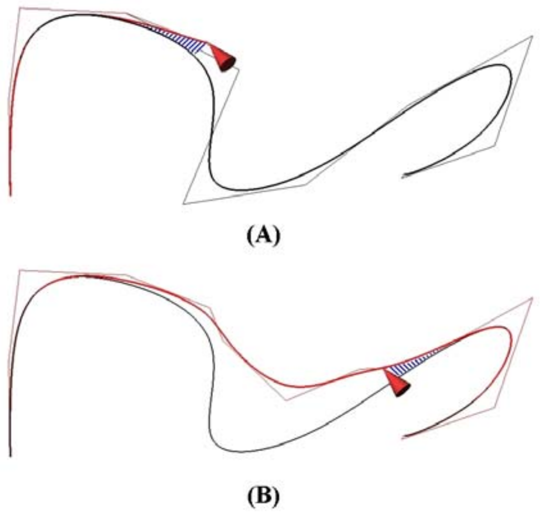


Fig. 12: The blue zone indicates where the new curve deviates from the old one

6- Conclusions and further work

In the paper the techniques for sketching and modifying curves and surfaces employed in VEIS have been discussed. The over-sketching technique has been emphasised because it particularly meets the requirements of conceptual free form modelling in virtual environment. In our opinion the over-sketching is really a fast and intuitive technique that the designer could use without "fighting" against the complexity of traditional CAD systems while continuing to employ the usual hand drawing gestures.

However VEIS is still far from being complete as a sketching system and needs further research in order to be able to manage complex shapes composed of many surface patches, sketch curves by interpolating technique [17], control the quality of the surface. In the near future it will be necessary to transfer VEIS to a more advanced VR system (power-wall, CAVE, virtual table) in order to set a benchmarks and test the system with designers.

7- Bibliography

[1] F. Bruno, M. Muzzupappa, P. Postorino and S. Rizzuti VEIS: Virtual Environment for Interactive Sketching. In 1st Eurographics Italian Chapter, Milano - Italy, 2002

[2] F. Bruno, M. L. Luchi, M. Muzzupappa and S. Rizzuti. A Virtual Reality desktop configuration for free-form surface sketching. In XIV INGEGRAF, Santander - Spain, 2002

[3] F. Bruno, M.L. Luchi, M. Muzzupappa and S. Rizzuti. Interactive Modification of free-form curves and surfaces in Virtual Environment. In XIII ADM – XV INGEGRAF, Napoli – Italy, 2003.

[4] E. Sachs, A. Roberts, and D. Stoops. 3-Draw. A tool for designing 3D shapes. In IEEE Computer Graphics and Applications, 11(6):18-25, 1991

[5] D. Liang and M. Green. JDCAD. A highly interactive 3D modelling system. In Computers and Graphics, 18(4):499-506, 1994

[6] J. Butterworth, A. Davidson, S. Hench, and T. M. Olano. 3DM: A three dimensional modeler using a head-mounted display. In Symposium on Interactive 3D Graphics, Cambridge - Massachusetts, 1992

[7] T.H. Dani and R. Gadh. COVIRDS A Conceptual Virtual Design System. In Computer Aided Design, 29(8):555-563, 1997

[8] A. Stork, O. Schimpke and R. De Amicis. Sketching Freeforms in semi-immersive virtual environments. In ASME 2000, Baltimore - Maryland, September 10-13, 2000.

[9] M. Fiorentino, R. de Amicis, G. Monno and A. Stork. Spacedesign: A Mixed Reality Workspace for Aesthetic Industrial Design. In IEEE and ACM International Symposium on Mixed and Augmented Reality, Darmstadt - Germany, September 30 - Oct. 1, 2002

[10] S. Schkolne, M. Pruett and P. Schröder. Surface Drawing: Creating Organic 3D Shapes with the Hand and Tangible Tools. In SIGCHI conference on Human factors in computing system. Seattle - Washington, 2001

[11] A. Liverani and G. Piraccini. Full-Scale Surface Modeling in Virtual Reality. In 12th ADM, Rimini - Italy, 2001

[12] C.G.C. van Dijk and A.A.C. Mayer. Sketch input for conceptual surface design. In Computers in Industry, 34(1):125-137, 1997

[13] J. P. Pernot , S. Guillet, J. C. Léon, C. E. Catalano and F. Giannini. Character line deformation feature. In Eurographics Italian Chapter, Milano - Italy, 2002

[14] E. Ferley, M.P. Cani and J.D. Gascuel. Virtual Sculpture. In Eurographics '99, Milano - Italy, 1999

[15] E. Ferley, M.P. Cani and J.D. Gascuel, Resolution Adaptive Volume Sculpting, In Graphical Models, march 2002

[16] B.P. Perles and J.M. Vance. Interactive Virtual Tools for Manipulating NURBS Surfaces in a Virtual Environemnt. In ASME 1999 Industrial Virtual Reality: Manufacturing and Design Tool for the Next Millenium Symposium. Chicago, Illinois, November 1999

[17] M.J. Banks and E. Cohen. Realtime spline curves from interactively sketched data. In Computer Graphics, 29(2):99-107, 1990

[18] R. De Amicis, F. Bruno, A. Stork and M. L. Luchi. The Eraser Pen: A New Interaction Paradigm for Curve Sketching in 3D. In Design 2002, Dubrovnik - Croatia, 2002

[19] L. Piegl and W. Tiller. The NURBS Book. Springer, 1995