A CAD SYSTEM IN AUGMENTED REALITY APPLICATION

Pier Paolo Valentini[(a)], Davide Gattamelata[(b)], Eugenio Pezzuti[(c)]

[(a)]Università degli Studi di Roma Tor Vergata
[(b)] Università degli Studi di Roma Tor Vergata
[(c)] Università degli Studi di Roma Tor Vergata

[(a)]Valentini@ing.uniroma2.it, [(b)]Gattamelata@ing.uniroma2.it, [(c)]pezzuti@ing.uniroma2.it

ABSTRACT

In this paper an integration between a computer aided 3D modeller and an augmented reality environment is presented. The system is based on an high resolution web cam to acquire video stream from the real world and an electromagnetic tracking system (Flock of Bird by Ascension) which allows the user to interact with real and virtual objects in the augmented scene. The software to manage user interaction and data flow is implemented in Visual C++ and it makes use of the Artoolkit libraries, the OpenGL libraries and the Flock of Birds libraries. The purpose of the system is to speed up reverse engineering and prototyping processes, because the user can relate real object features in the scene to model its virtual entities or acquire geometrical features of existing parts. Moreover, the user can export the virtual models into CAD system or import external models to see how they fit in their real environment.

Keywords: Augmented Reality, CAD, motion tracking

1. INTRODUCTION

The Augmented Reality (AR) is an emerging field which deals with the combination of real world image and computer generated data. With an AR system, the user can extend the visual perception of the world, being supported by additional information and virtual objects. At present, most AR research is concerned with the use of live video imagery which is digitally processed and "augmented" by the addition of computer generated graphics. Many engineering applications are based on the video see-through system that is based on the use of one or two cameras which acquire an image stream from the real world. This stream is processed by a computer which produces an augmented image stream which is projected again to the user by means of a blind visor. The major aim of an augmented reality (AR) system is the need of a consistent registration between the real world and the virtual one, making the illusion that the virtual objects are part of the real scene (Bimber and Raskar, 2005). On the other hand, many scientists and technicians of the augmented reality pay attention on the integration process of virtual features in real scene without taking care of interactivity problems. Referring to some AR applications as the maintenance of aeronautical or mechanical equipments or surgical applications, the lack of interactivity seems not to be a problem. Indeed, for these purposes, the cheapness of an AR system to visualize mixed scenes, with low cost of implementation, appears the winning choice.

Moving the attention to the industrial field, according to many researchers, the AR systems seem to be the future of Computer Aided Design (CAD) technologies and Virtual Engineering. The idea is to develop an AR system which is able to support the design in engineering and science. The first step towards this target is that the designer has to be able to model the shapes of products in the augmented world, evaluating real time their interaction with the real world. This means that the user not only has to perceive the augmented environment in a visual way, but also has to interact with it communicating his ideas. For this purpose, the visual scene has to be provided with an additional tracking system that is able to record the position of a virtual pen (the pointer) that is used to model the shapes (Liverani, Amati, Carbone, Caligana, 2005). Scientific literature reports the usage of mainly two types of tracking devices: optical trackers (Fiorentino, 2003) and haptic system (Bruno, Caruso, Pina, Muzzupappa, 2007), (Vallino, 1998). The usage of an haptic device to track the position of a pointer in the scene produces extremely precise results, but the working space reached by the robotized arm is limited. In addition the movement of the user is not completely free. The usage of optical systems allows a wider working area with less precision. It is also affected by illumination problem and marker visibility requirements.

An adequate trade off among precision, working space and user friendliness is the use of magnetic tracking devices. For the investigation presented in this paper the authors have tested the *Flock of Birds* by Ascension (Ascension Technology Corporation, 2002).

The data from the tracking system have to be real time combined with the camera video flow. Due to the registration process the computer can align the CAD objects with the user point of view. At the same time the pc records the user's events like position of the pointer or CAD command activation. So the user is able to model virtual objects moving all around the scene.

2. SYSTEM CO NFIGURATION AND SETUP

The implemented system (Figure 1) is made of input, processing and output devices. The input devices are a USB2.0 camera (Microsoft LifeCame VX-6000), the motion tracking system *Flock of Birds* (further details at: http://www.ascension-tech.com/products/flockofbirds.php) and a common pc keyboard. The first one supplies an artificial vision of the scene while the second provides the position of the pointer in the scene. The keyboard is used for key commands. The processing unit is a pc (Intel Pentium 4, CPU freq. 2,8GHz, motherboard Asus P5 series and graphic card NVIDIA 6600 series).

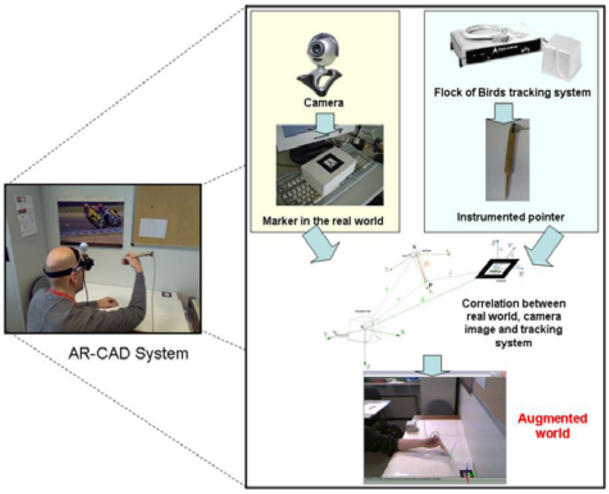

Figure 1: Overview of the implemented system.

The output device is a video see-trough display (eMagin Z800 3DVisor, Aspect ratio 4x3, Resolution SVGA 800 x 600 triad pixels per display (1.44 megapixels), contrast ratio >200:1, brightness >50cd/m2, 24 bit color, headtracking 360° horizontal, >60° vertical). The pc processes the input data and combines them to obtain the video data flow of the augmented scene.

The data flows coming from the input devices have to be synchronized and collimated by means of two relations: one between the real world and the camera and the other between the real world and the tracking system. The collimation between the virtual world and the real one involves some mathematical transformations. It is necessary to express the pointer tip position (*P* in Figure 2) in the reference frame attached to the marker and then apply the camera transformation to project the point according to the camera point of view.

With reference to Figure 2, the first transformation concerns the evaluation of position of *P* in the reference frame of the transmitter (O-*XYZ*) and it can be written as:

$$\{r\} = \{s\} + \left[M\left(\Psi,\theta,\varphi\right)\right]\cdot\{P_t\} \tag{1}$$

where $\{r\}$ contains the coordinates of the tip *P* in the transmitter reference frame (*S-XYZ*); $\left[M\left(\Psi,\theta,\varphi\right)\right]$ is the rotation matrix built from the attitude angles $\left(\Psi,\theta,\varphi\right)$ of the sensor reference frame (S-$x_s y_s z_s$); $\{s\}$ is the translation vector from the origin of the transmitter reference frame to that of the sensor; $\{P_t\}$ is the vector which indicates the local position of the tip *P* in the sensor reference frame.

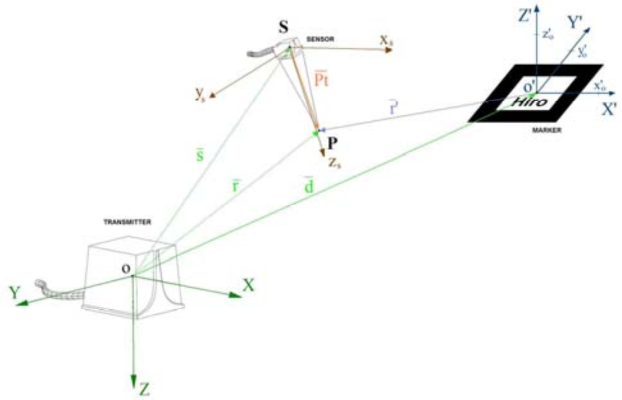

Figure 2: The collimation Process.

The second transformation concerns the computation of the coordinates $\{r'\}$ of pointer tip *P* in the reference frame of the marker (*o'-X'Y'Z'*). This transformation can be written as:

$$\{r'\} = -\{d\} + \left[T\left(\Psi',\theta',\varphi'\right)\right]\cdot\{r\} \tag{2}$$

where $\{d\}$ is the vector between the origin of the marker reference frame an that of the transmitter; $\left[T\left(\Psi',\theta',\varphi'\right)\right]$ is the rotation matrix between the marker and the transmitter reference frames. The distance $\{d\}$ and the matrix $\left[T\left(\Psi',\theta',\varphi'\right)\right]$ are independent from the location of the pointer in the scene and have to be computed only at the beginning of the registration.

The registration process can be made by the user selecting with the pointer tip *P* the origin *O'* and two points on X', Y' axes of the reference frame of the marker.

In order to project the virtual object in the scene another transformation has to be applied. This transformation involves the computation of the relative position between marker and camera. It concerns the recognition of the marker features by pattern matching (Figure 3) and the computation of a distance vector and a rotation matrix.

These two entities can be used to render any virtual object (augmented contents) on the scene using the graphic pipeline typical of OperGL (Wright, Lipchak, 2004) (Figure 4).

The camera transformation is based on the calibration procedure, which aims to know the intrinsic and extrinsic parameters of the camera. They are necessary to link the pixel coordinates of an image point to the corresponding coordinates in the reference frame of the marker (Trucco, Verri, 1998) and it is supplied by the ARToolkit libraries (Lamb, 1999).

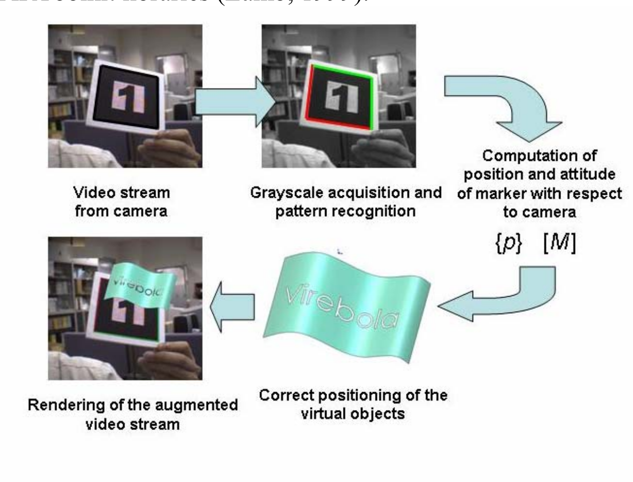

Figure 3. Pattern recognition pipeline.

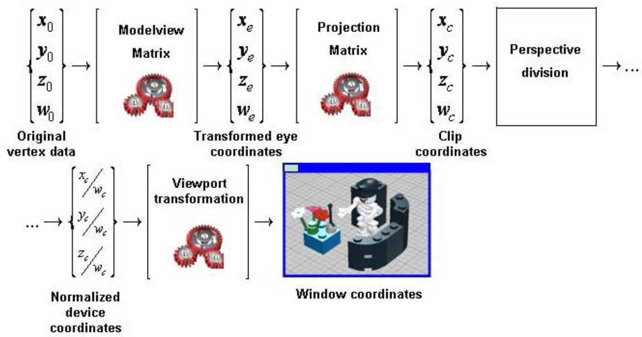

Figure 4: The render transformations pipeline.

3. DATA FLOW AND CAD ENGINE

The core of the proposed system is the *ad hoc* developed software AR-CAD 1.0. Its role is to manage the data flow coming from camera and sensor and to process the output video stream. Mainly there are three different data flows in the system (see Figure 5).

The first data flow comes from the camera and interests the image processing section of the software. Using the artoolkit method *ARGetTransMat()* the software computes the camera parameters transformation described in the previous section of the paper.

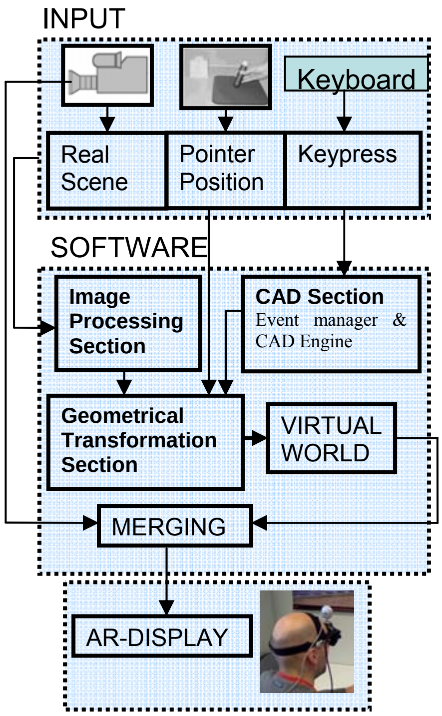

Figure 5 : Data Flow Scheme.

The second processes the data flow coming from the *Flock of Birds* and uses the geometrical transformation section already presented. In this way the equations (1) and (2) are used to compute the pointer tip coordinates in the reference frame of the marker.

The third data flow concerns the user's key-press events and is processed in the CAD section.

The CAD section is made of two parts: the modelling core and the events manager (see Figure 5). The modelling core is a set of classes and methods which are used to store and manipulate the virtual objects. The class *cAR_CAD* manages every geometrical entities (curves, lines, planes, surfaces and solid) in the 3D virtual space. When the user activates a CAD function, the event manager uses the corresponding methods of the library (*AR_Functions.h*) to generate an instance of the *cAR_CAD* class and to store the selected points in a right way.

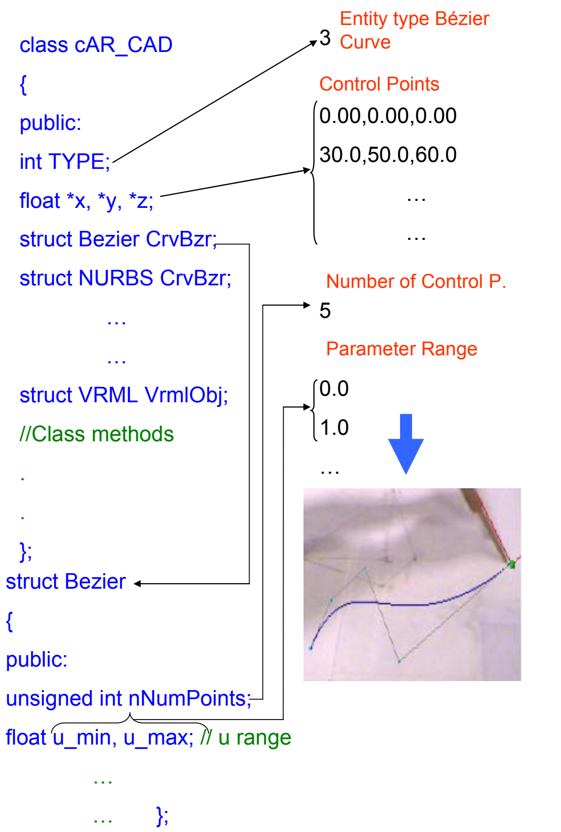

Figura 6: An example of how to use the cAR_CAD class to manage a Bézier curve.

Referring to the structure (Figure 6) each object is defined by an integer (named TYPE) which represents the type (line, polyline, parametric curve), then by three dynamically allocable arrays *x,y,z* to store the list of points and finally by the relative data structures. Depending on the entity to build, only one of the data structures is initialized. Parametric curves and parametric surfaces (like NURBS or Bézier curves and surfaces) have dedicated data structures. The solid entities use a common structure which is similar to the VRML one (Hartman and Wernecke, 1998).

Finally all the data flows converge in the virtual world section (see Figure 5) which elaborates the objects in the memory stack to project them in the augmented scene. The last operation is that of merging, in which the software render the virtual objects to the real world and send the augmented video frame to the AR-Display.

Two classes of objects have been implemented. The first one comprises the entities stored using only one instance of structure (line, polyline, parametric curves and surfaces by points selection). The second includes the entities stored with nested data structures and is used for building solid objects and surfaces by extrusion, sweep or loft.

With reference to Figure 7, the structure of the solid is defined starting from the basis sketch (green hexagon), which is stored in a curve data structure type.

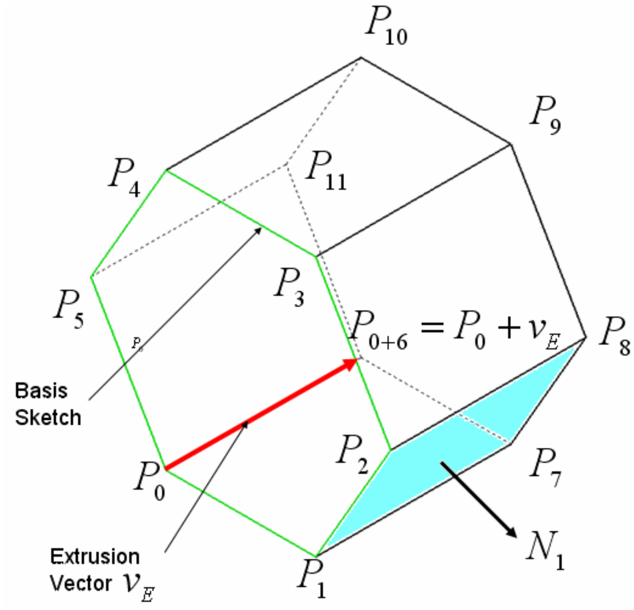

Figure 7: An example of nested data structures to build an extruded solid.

Following the *VRML* syntax in order to index the faces, the vertices of the extruded face are ordered in sequence. For an extrusion of a polygon of n vertices $(P_0,\ldots,P_{n-1})$ we have a solid object of $2n$ vertices $(P_0,\ldots,P_{n-1},P_n,\ldots,P_{2n-1})$ with $n+2$ faces. The secondary vertices are calculated through the following expression:

$$P_{i+n} = P_i + \overrightarrow{v_E} \qquad (i = 0,\ldots,n-1) \tag{3}$$

where $\overrightarrow{v_E}$ is the extrusion vector.

In order to draw the stored entities in the scene, the software uses the functions implemented in *AR_Functions.h*. These functions make use of the basic OpenGl methods *glBegin()* and *glVertex*`3f` using the mode *GL_LINE_STRIP*, to draw curve pieces or lines, and the *GL_POLYGON* to draw faces and tessellated surfaces [R. S. Wright et al. 2004].

Looking at the extrusion example of Figure 7, the graphic engine draws $n+2$ faces F_i $(i = 0..n+1)$. F_0 is the face of basis sketch and depends on the basis sketch vertices $(P_0,\ldots,P_{n-1})$ and F_{n+1} is the extruded face and depends on the list of secondary vertices $(P_n,\ldots,P_{2n-1})$. At last the n lateral faces F_k $(k = 1..n)$ which depend on the lists of four vertices $(P_k, P_{k-1}, P_{k-1+n}, P_{k+n})$. The vertices of lateral faces are ordered counterclockwise to have the face normal pointing outside the solid (see N_1 in Figure 8).

4. MODELING IN AUGMENTED REALITY

The geometric modeler in the AR-system has been implemented as a feature based builder, as it happens in many recent CAD programs. For this purpose, it is possible to classify the implemented functions in two categories: that of the sketch commands and that of the applied functions.

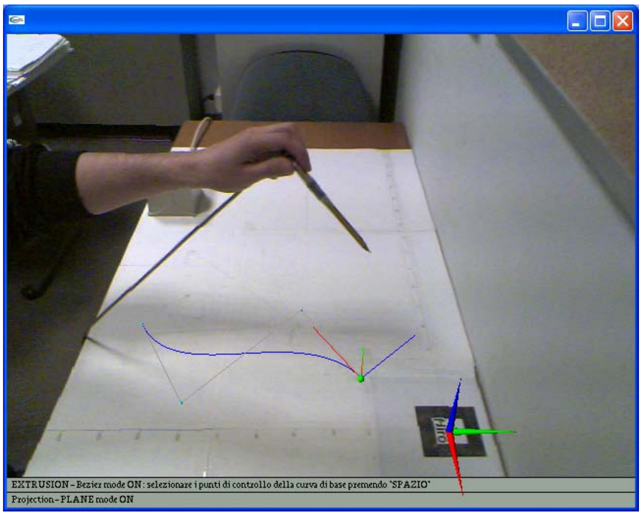

Figure 8: Drawing of parametric Bézier Curve projected on the selected sketch plane.

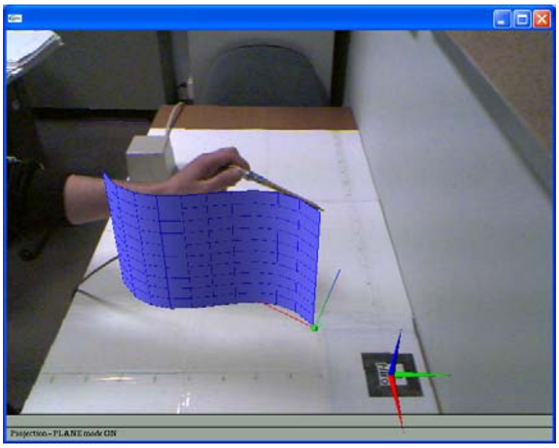

Figure 9: Drawing of a surface by mean of curve extrusion

In the first category, we have elementary geometrical shapes and parametric plane curves. The **Figure** 8 reports the drawing action of a parametric curve projected on a sketch plane.

The category of applied functions to the sketch is composed from two subclasses, the class for surface construction and the class for solid modeling.

Figure 9 reports an example of surface construction by means of an extrusion of the Bézier curve in the previous example.

Solid objects can be built selecting a sketch and applying a specific function. **Figure** 10 reports two cases: an extrusion of a circle to model a cylinder and that of a closed profile.

The 3D entities can be also built selecting points in space without the use of sketch plane. An example of solid modeling of a sphere is reported in Figure 11 while an example of a generic surface is reported in Figure 12.

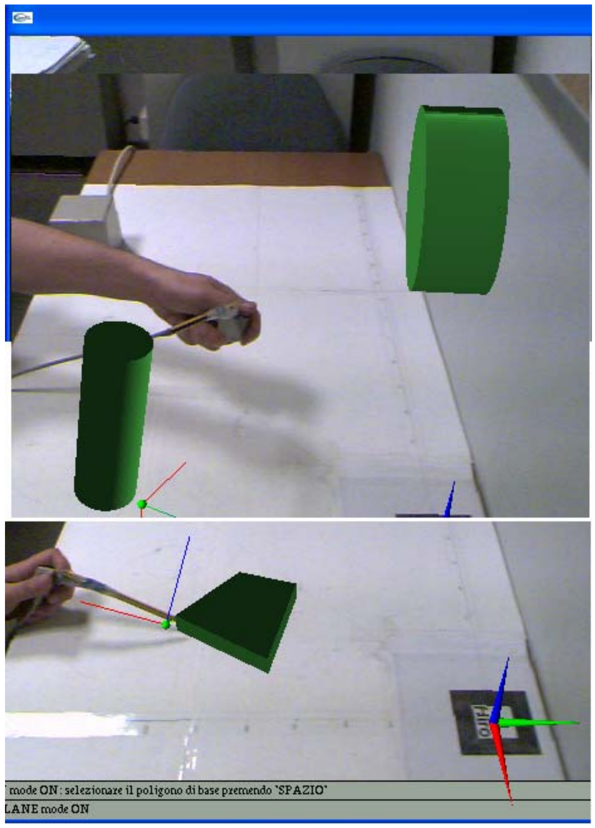

Figure 10: Solid parts from interactive extrusion (two cylinders, on the top; a generic polygon extrusion, at the bottom).

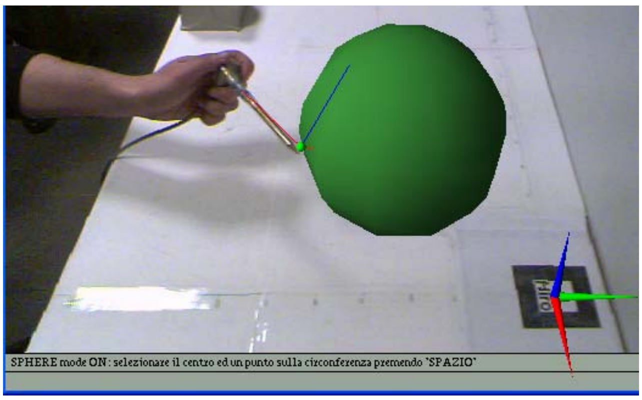

Figure 11 Solid modeling of a sphere.

The major advantages of drawing entities referring directly to real points and real objects can be noted for the complex surface construction by means of points selection. (Figure 12).

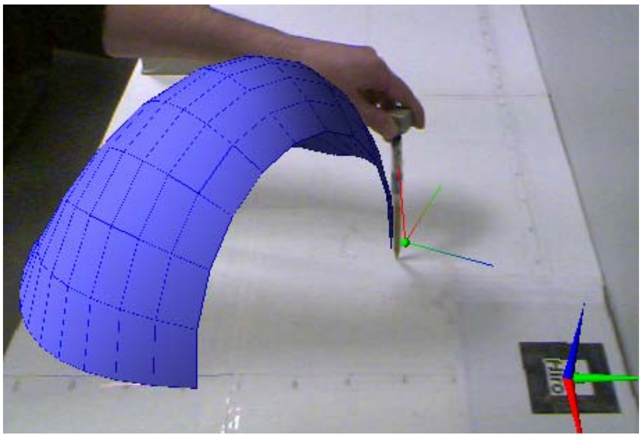

Figure 12 Surface Modeling.

5. CONCLUSIONS

In this paper an hardware and software setup has been presented in order to implement a modeling environment in augmented reality. The described architecture is able to speed up reverse engineering and prototyping tasks at a reasonable cost and with a good reliability and precision. The great advantage for the designer is that he can model and modify objects using an intuitive device (the magnetic pointer) and directly in the real world. The combined use of camera, pointer and head mounted displays have been managed by means of a specific software. This software play the role of an interface between the hardware devices (input and output) and the graphics engine. The modeling procedures are based on the OpenGL libraries. The use of a magnetic pointer has the advantage to widen the modeling space and make the picking operations simpler.

For engineering and industrial applications, one of the crucial aspects is the picking precision of the instrumented pointer. It is mainly affected by two factors. The first depends on the intrinsic instrument precision and software implementation. The pointer position is sensitive to the static resolution of the motion tracking system (about 0.5 mm for the position and about 0.1° for the orientation) and the acquisition sampling rate. The *FOB* is able to acquire up to 144 frames per second (fps), but it is forced to work at 30 fps, synchronized with the camera. A lower sampling rate is also justified considering the limited velocity of user's movements during modeling.

The second contribution to the system precision depends on user's limited accuracy during picking, which is subjective.

Moving the attention to the AR component of the system some improvements can be introduced for future work. The code concerning the modeling engine can be extended with new commands to speed up the modeling procedure. Moreover, the implementation of functions for advanced surface and solid editing can improve the modeling capabilities similarly to modern CAD environments.

Figure 13: Reverse engineering of a toy car roof (on the top) and of a chocolate box.

In addition, the interactivity of the system can be improved introducing other devices like virtual glove or introducing a virtual keyboard. At last, to increase the illusion of the augmented scene, it is desirable to have a collimation between the illumination of the real scene and that of the virtual one.

REFERENCES

Ascension Technology Corporation, 2002. *Flock of Birds API reference guide.* Available from: ftp://ftp.ascensiontech.com/DRIVERS/WINDOWS_DRIVER/ [accessed 10 June 2008].

Bimber, O. and Raskar, R., 2005. *Spatial Augmented Reality Merging Real and Virtual Worlds 2005.* United States of America: A K Peters, Ltd.

Bruno, F., Caruso, F., Pina, M., Muzzupappa, M., 2007. Sviluppo di un tool di augmented reality di supporto alle tecniche di Reverse Engineering per contatto. *Proceedings of ADM Congress*. 6-8 Giugno 2007, Perugia (Italy).

Fiorentino, M., 2003. *Design estetico in mixed reality*. Phd Thesis, Politecnico di Bari, Italy.

Hartman, J. and Wernecke, J., 1998. *The VRML 2.0 Handbook*, United States of America: Silicon Graphics Inc.

Lamb, P., 1999, *Artoolkit.* Available from: http://www.hitl.washington.edu/artoolkit/ [accessed 10 June 2008]

Liverani, A., Amati, G., Carbone, L.,G. Caligana, G., 2005. Modellazione e ricostruzione video assistita di componenti. *Congresso congiunto ADM INGEGRAF*, Giugno 2005, Siviglia (Spain).

Trucco, E., Verri, A., 1998. *Introductory Techniques for 3-D Computer Vision*, New Jersey: Prentice Hall.

Vallino, J. R. 1998. *Interactive Augmented reality*. Phd Thesis, University of Rochester, New York.

Wright, R. S., Lipchak, B., 2004. *OpenGl Super Bible, third edition*. United States of America: Sams Publishing