

SKETCH-BASED INTERFACE FOR 3D PARAMETRIC MODELLING

F. Naya Sanchis^(s), M. Contero González, N. Aleixos Borrás, P. Company Calleja

Abstract

In this paper we present an experimental prototype called GEGROSS (GEsture & Geometric ReconstructiOn based Sketch System). This application allows using freehand sketches to create 3D parametric models onto a Tablet-PC. The modelling process consists of two stages. First, the user introduces the geometry of the 2D section using continuous sketches composed by line segments, circles and arcs, and then he can add, delete or edit the dimensional and geometric constraints by means a gesture alphabet (a gesture in this context represents a graphical symbol that is translated into a command). The supported gestures are geometry (line, arc, circle or ellipse), or constraint (dimension, parallel, perpendicular, tangent, concentric, horizontal or vertical). Later, in the second stage and also using gestures it is possible to make an extrusion or a revolution of the parametric section generated in the previous stage to create a 3D model. Then this process can continue, sketching new 2D sections onto the faces of the generated object and applying the corresponding modelling gestures.

The application uses the geometric kernel ACIS to store geometric entities, and the 2D DCM constraint manager from UGS D-Cubed firm to manage geometric and dimensional constraints of the parametric sections.

Preliminary tests have concluded the GEGROSS application is a feasible alternative to current approach used in commercial CAD applications in order to create shapes of small or medium complexity. In that situation it presents a more effective modelling time, and it has been rated as easier to learn than comparable commercial applications.

Keywords: Sketch-based interfaces, parametric modelling, gestural modelling, constraint based design, product design

1. Introduction

Sketching is the natural communication technique widely used by designers and engineers and is a useful and a powerful tool that helps designers during the fixation of new products (Tversky, 2002), because greatly enhances design creativity. For that reason, the “paper and pencil” approach is the basic support and is better suited for expressing the designer creative ideas at the early stages of product design process in a fast way.

The question is how sketching is supported by current CAD systems? Current commercial CAD systems are not specially, tailored to support sketch-based design. In this context, the problem is that this sketch-based design continues to be unplugged to the rest of the design process, and after the conceptual design is over and a final sketch has been obtained, the designer must create the CAD model from scratch.

In other words, in spite of recent advances in Computer Aided Design and large strides made by commercial CAD applications in terms of descriptive power, flexibility and functionality, current CAD tools are not well suited to the initial design stages of product development, because many techniques and idioms characteristic of hand-made drawings cannot be used

directly in CAD systems and it is still difficult to capture the drawing language used in conceptual design, where designers prefer pencil and paper over computer-based tools. Available Graphical User Interfaces (GUI) for CAD applications are still by and large constrained by the WIMP (Windows, Icons, Menus and Pointing) paradigm, which severely hinders both ease of learning and ease of use. In this case, users are forced to learn a very different methodology involving the use of menus combined with certain keyboard commands or mouse actions. Present-day interfaces require designers to leap a large conceptual gap from their image of a desired object to the geometric model that formally expresses its shape. To sum up, there is a disconnection between sketching and CAD tools in the new product development process, because current CAD applications are not suitable tools to integrate both conceptual design (where sketches dominate) and detailed design (where three-dimensional modelling is the nucleus).

Therefore, true computer-aided sketching (CAS) tools, oriented towards design, are required. Sketch-based modelling systems must facilitate the speedy creation of detailed 3D designs, usually from elementary 2D sketches, and to improve support for the flexible generation of alternative design ideas at the early stages of design (Cera et al, 2002), where non-formalized and non-structured ideation processes could flow freely and final 3D shapes would be automatically derived from such ideation processes.

In this paper we try to show one of our developments applications to support sketching with computers. The aim of these works is to explore new interaction paradigms in CAS tools, geared at exploiting sketching skills of designers and engineers.

2. Sketch-Based Interface and Modelling (SBIM)

During last decades different research lines has been explored to improve the human-computer interface in CAD systems. Sketch-based interfaces and modelling (SBIM) is an emerging research field, and seems to be able to create the computer tools required to shift (Igarashi and Zeleznik, 2007) to a new paradigm where sketches should be used as input to create digital engineering models in a true computer-aided ideation environment. Recent advances in the field of SBIM promise better integration of those separated worlds (sketching and CAD tools). Work on SBIM has looked at a paradigm shift to change the way geometric modelling applications are built, in order to focus on user-centric systems, rather than systems that are organized around the details of geometry representation. While most of the activity in this area in the past has been focused in off-line algorithms, the growing focus on sketches and modelling has brought forth a new emphasis on approaches geared towards interactive applications. These interactive applications use a digitizing tablet and a pen, an approach termed calligraphic interfaces (see *Computers & Graphics* vol. 24, special issue "Calligraphic Interfaces: towards a new generation of interactive systems"). These rely on interactive input of drawings as vector information (pen-strokes) and gestures, possibly coupled with other interaction modalities. In these interfaces the artificial dialogue constraints imposed by the previous generation of user interfaces are removed and designers can work with the computer much the same as they would with more traditional media, to capture rough shapes and ideas. This new generation of calligraphic applications uses gestures and pen-input as commands (Fonseca and Jorge, 2001). The development of these new graphical user interfaces opens new perspectives to the creation of tools oriented to satisfying the needs of the designer in conceptual design stage. This is in contrast to the current generation of systems, which seem to take the approach of billing themselves as Application Programming Interfaces (APIs) to geometry kernels, which do make a poor job of hiding the internal details of the underlying representations.

To sum up, in contrast to conventional drawing applications, the stylus can also be used to enter continuous-mode sketches and freehand strokes. Thus, there is a growing research

interest in using freehand drawings and sketches as a way to create and edit three-dimensional geometric models. Digital sketching must replace paper-and-pencil sketching, and it must become a “natural” process does not disturb the user. It seems feasible now, since the reasons for its absence come only from the facts that appropriate hardware was not available until the recent advent of Tablet PCs.

These advanced CAS systems should provide more functionality than paper or a whiteboard, trying to provide an added value to sketching on a digital environment. This extra functionality has been directed to improve the graphic quality of the sketch by means of beautification functionality, providing as output an improved 2D representation, or it has been oriented to automatically transform the 2D sketch into a 3D model. So, it must help the designer to turn the final idea into input for the following designing phase. Intelligent support for automatically recognizing the sketch is the way to integrate sketching into standard design applications; hence this is the advanced functionality final goal. This advanced functionality justifies the importance of automatic creation of models from sketches.

Within this research area it is possible to distinguish two principal approaches to transform the 2D sketch into a 3D model in sketch-based modelling. One method relies on gesture alphabets as commands for generating objects from sketches (a gesture in this context represents a graphical symbol that is translated into a command). Examples of gestural systems are SKETCH (Zelevnik et al., 1996), Teddy (Igarashi et al., 1999), GIDeS (Pereira et al., 2000) and Blobmaker (De Araujo and Jorge, 2003). Gestural systems provide predefined gesture alphabets that encode some geometric modelling operations; basically these systems substitute the selection of icons and menus by graphic gestures. The second approach, derived from computer vision, uses algorithms to reconstruct geometric objects from sketches that depict their two-dimensional projection. Examples of reconstruction systems are Stilton (Schweikardt and Gross, 2000), Digital Clay (Turner et al., 2000) and CIGRO (Contero et al., 2005). Reconstruction modelling is the alternative to gestural modelling, where geometric reconstruction techniques apply to build the object’s geometry from a sketch that represents the object’s projection.

In summary, it seems to be a great consensus on pursuing freehand sketching and system detection of designer intents. However, two alternatives exist to capture those designer intents: reconstruction based and gesture based. From these approaches the reconstruction based is the most transparent to the user, since he has only to create the sketch and it doesn’t require a priori knowledge of a gestural command set. Chronologically reconstruction systems appeared before gestural ones, because reconstruction systems took advantage of previous works in the offline line drawing recognition field. Gestural systems require more elaborate recognition engines for distinguishing geometry information and gestural codes, although the first systems avoided this using icons and menus to explicitly give this information to the system.

3. The GEGROSS application: System Operation

The prototype GEGROSS (GEsture & Geometric ReconstructiOn based Sketch System) is a CAS application than performs an online conversion of a raw sketch into a three-dimensional model. The GEGROSS application is a gestural system and allows the user to generate three-dimensional models, using some gestural commands. In this system, it is possible to draw two-dimensional parametric freehand sections combined with the use of a simple gesture alphabet that encode some geometric modelling operations (extrusion and revolution).

The gestural modelling process is organized in two stages. In first place a 2D profile is defined, and then using some of the modelling gestures, new geometry is building. To do this, the user introduces the geometry of the 2D section using a two-dimensional parametric

freehand sketch module of GEGROSS, called ParSketch (Naya et al., 2007a). This module implements a reduced-instruction set calligraphic interface paradigm to manage the geometric constraints found in two-dimensional sections. If user wants to generate design alternatives, or adjusting some sketch to reach some dimensional condition, the system provides parametric capabilities and handwritten dimensional control to the two-dimensional freehand sections. The system offers a very simple interface to create parametric sections with an interesting possibility: “dimensional control”. In this way, user can impose some dimensional condition drawing the corresponding dimension and writing its value. Handwritten text recognition capabilities provide a very simple method to change dimension values. Later, in the second stage and also using gestures it is possible to make an extrusion or a revolution of the parametric two-dimensional section generated in the previous stage to create a 3D solid model. Then this process can continue sketching new 2D sections onto the faces of the generated object and applying the corresponding modelling gestures.

As can be seen in Figure 1, the developed application allows the user to introduce the freehand sketch directly onto a Tablet PC, using the reduced-instruction set calligraphic interface.



Figure 1. User stroke input on a Tablet PC (GEGROSS)

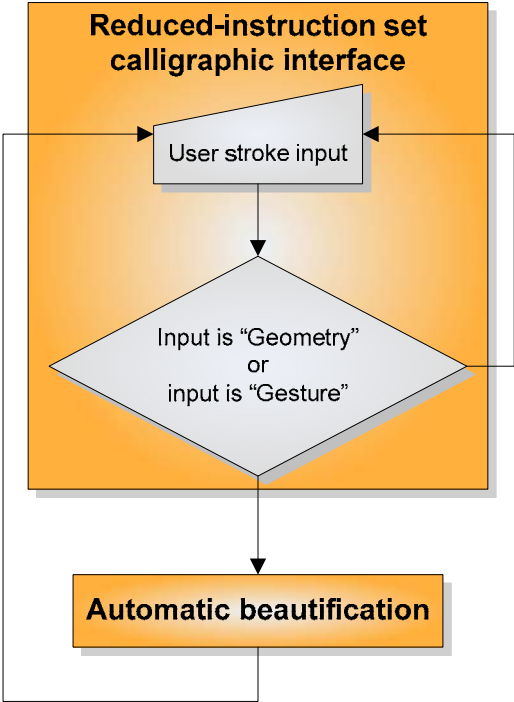


Figure 2. Stages of the prototype system GEGROSS

In Figure 2 the structure of the system is presented. When the user introduces a new stroke, the system uses the drawing pressure as a mode discriminator (geometry or gesture). Then, the application recognizes the type of stroke drawn by the user using a geometry recognizer (RecoGeo) and a gestural recognizer (RecoGes), neglecting the information which is not necessary. Next, automatic beautification stage is executed in sequence. In this way, the parametric drawing is always updated. The geometric recognizer RecoGeo has been developed to analyze and convert a sketch into its constituent primitive, allowing users to build simple sketches composed by line segments and arcs, which are automatically tidied and beautified. Once the designer has introduced the complete outlined sketch, it can be edited, dimensioned and constrained using the gesture recognizer RecoGes. RecoGes has been developed to provide an alphabet of geometric/dimensional constraints to parameterise the sketches and for classifying a gesture alphabet to define commands for modelling operations of solid models. Real-time sketching is supported because it integrates a reduced-instruction set calligraphic interface, an automatic beautification, the geometry recognizer (RecoGeo) and the gestural recognizer (RecoGes).

The prototype system use the geometric kernel ACIS (www.spatial.com) to store the geometric entities, and the 2D DCM constraint manager from the UGS D-Cubed firm (www.ugs.com) to manage the geometric and dimensional constraints of the drawings. Handwritten number recognition is provided by the Windows XP Tablet PC Edition operative system.

3.1 Reduced-instruction set calligraphic interface

With regards to the reduced-instruction set calligraphic interface, the design goal of this interface is to create two-dimensional parametric sections and three-dimensional models in a very simple way, using the conventions of technical drawing to define the shape of the section. Therefore, the system is a first step to evaluate the feasibility of using a sketching interface to define two-dimensional parametric sections and three-dimensional parametric objects, and implements a gesture alphabet and a calligraphic interface to manage the geometric constraints found in two-dimensional sections.

The user interface is responsible for low-level user-application interaction and its functionality is designed to minimize (minimalist interface) the interaction with menus or icons in an attempt to emulate the traditional use of pen and paper employing drawing conventions used in technical drawings. In other words, the main objective that guided interface design for this system was to provide an intuitive drawing strategy that takes into account, the way engineers create sketches and technical drawings. In this reduced-instruction set calligraphic interface, the user concentrates in sketching instead having to navigate through menus or searching for icons in toolbars.

The user introduces the freehand sketch directly on the Tablet-PC screen. The points of the stroke are taken by means of the Wintab API (www.pointing.com), an open interface that directly collects pointing input from a digitizing tablet and passes it to applications in a standardized fashion. This API allows retrieving additional information as the pressure the user applies at each point of the stroke over the tablet. Raw strokes are then processed by the RecoGeo recognizer or by the RecoGes recognizer. In this way, the application analyzes the typology of the sketched entity, and if it corresponds to a geometric shape or to a gestural command, the entity is processed, otherwise it is ignored. The application then waits for a new data input stroke from the user.

In the reduced-instruction set calligraphic interface developed, the command set includes just two main concepts:

- Geometric strokes: they allow designers to create geometric entities to build a section gradually.

- Gestural alphabet strokes: they allows users to selectively add o edit parametric constraints of the entities to possibly define more complex and well defined sections, and allows users to generate three-dimensional objects (Table 1).














Constraint gestures	Class
	Concentric
	Dimension
	Diametral dimension
	Radial dimension
	Tangency
	Vertical
	Horizontal
	Parallel
	Perpendicular
	Cross-out (erase)
	Extrusion
	Revolve-right
	Revolve-left

Table 1. Gesture alphabet implemented in GEGROSS.

Regarding geometric strokes, GEGROSS supports sketched segments, which can be recognized as a geometric form of the type line, arc, circle or ellipse, or composed entities that are automatically segmented into those basic entities. As explained before, the mode detection has been solved using the electronic pen pressure information, since the system is intended to be used by persons with basic engineering drawing skills. It can be said that line width is the mode-change feature, when reading an engineering drawing. In the usual practice, thick lines are associated to geometry and thin lines to dimensions and other types of annotations. As line width is related to increasing pressure with the pencil while drawing, this information is used to discriminate among geometry or gesture. Therefore, a sketch “logic” condition that is imposed to define the sketching procedure was using pressure level as the main criterion to distinguish between real geometry primitives (line segments, arcs, circles and ellipses) and other information (construction lines, dimensions, edition gestures, geometric constraints ...). In other words, drawing making high pressure on the screen is intended for geometry input, while soft pressure is associated to auxiliary information. In this case, the application takes into account the pressure applied by the pencil, using a pressure level threshold configured by the user to classify strokes as geometry or gestures.

Examples of interaction with the developed interface of the application are presented in Figure 3, in Figure 4, in Figure 5 and in Figure 6.

As has been mentioned before, modelling process is organized in two stages. In first place a 2D profile is defined. One single stroke is accepted as input, and it is later decomposed into connected entities. The geometry is then beautified, and when the application shows the beautified version (Figure 3.b) the user can add another stroke. Once the designer has introduced the complete outlined sketch, it can be edited, dimensioned, constrained and lastly swept to create a solid. In other words, the interface offers some innovative ways of controlling the shape after a beautified constrained model is presented to the user. To do this, the system interprets user gestures as dimensional/geometric constraints (dimension, parallel, perpendicular, tangent, concentric, horizontal or vertical) to parameterize the sketch or as scratching gesture to edit the sketch. As can be seen in Table 1, sketches drawn by engineers are made with typical symbols of technical drawings, which have to be drawn in a typical way. That is because engineers are taught to introduce geometry and dimensions in a specific way. In other words, some multistroke entities must be drawn sequentially in time using a “logic” order. For instance, regarding the linear dimension gesture, extension lines are introduced first, in second place the dimension line is introduced and, lastly, left and right arrows are added respectively.

This optional stage for section definition is intended to provide geometric and dimensional control over the sketch. Here it is possible distinguish “automatic” and “user defined” constraints and dimensions. Automatic constraints and dimensions are provided by the system. The “user defined” ones are sketched by user. They can add new constraints drawing their associated gestures (Table 1) near the geometric entities where they must be applied. These gestures can be written by the user to impose some desired constraint. Also, using the scratch gesture, drawing entities can be removed. This not only allows errors to be corrected, but also enables more complex shapes to be drawn incrementally, by refining simpler forms as illustrated in Figure 3. When the user draws a scratch gesture, the application detects the entities that the user wants to delete as being those intersecting the smallest quadrilateral enclosing the scratching gesture. In Figure 3.e it is possible to see the use of the scratching gesture to refine the geometry. Then a horizontal constraint is applied by simply sketching its associated gesture over the segment the user wants to make horizontal, and a parallel constraint is applied by sketching its associated gesture over the two segments the user wants to make parallel (see Figure 3.h, Figure 3.i, Figure 3.j). These entities are drawn by applying low pressure with the stylus. Once the desired shape has been obtained, it is possible to proceed with dimensional control using the dimension gesture. If there isn't the dimension, a first action is to draw a dimension without the dimension text. This is interpreted by the application as a measure command, and the current value of that dimension is shown. If the user wants to change the current dimension value, he or she writes the new value next to the current one. Then the system regenerates and displays the new geometry (Figure 3.k). In this way, the system provides a very natural form of imposing the desired dimensions over the sketch.

Later, in the second stage and also using gestures it is possible to make an extrusion or a revolution of these parametric sections generated in the previous stage to create a 3D model. Then this process can continue, sketching new 2D sections onto the faces of the generated object and applying the corresponding modelling gestures. Examples of modelling with the GEGROSS application are presented in Figure 4, in Figure 5 and Figure 6.

3.2 Automatic beautification process

Concerning the automatic beautification process (automatic constraints and dimensions), it is in charge of adjusting the input sketch in real time in order to implement a true interactive sketching system. This stage performs an on-line conversion of recognized sketched entities into parameterized entities. Therefore, this online drawing beautification process provides an immediate feedback to the user, because it operates as the user draws the sketch. In this

manner, the section is adequately constrained and dimensioned at any stage of the section creation.

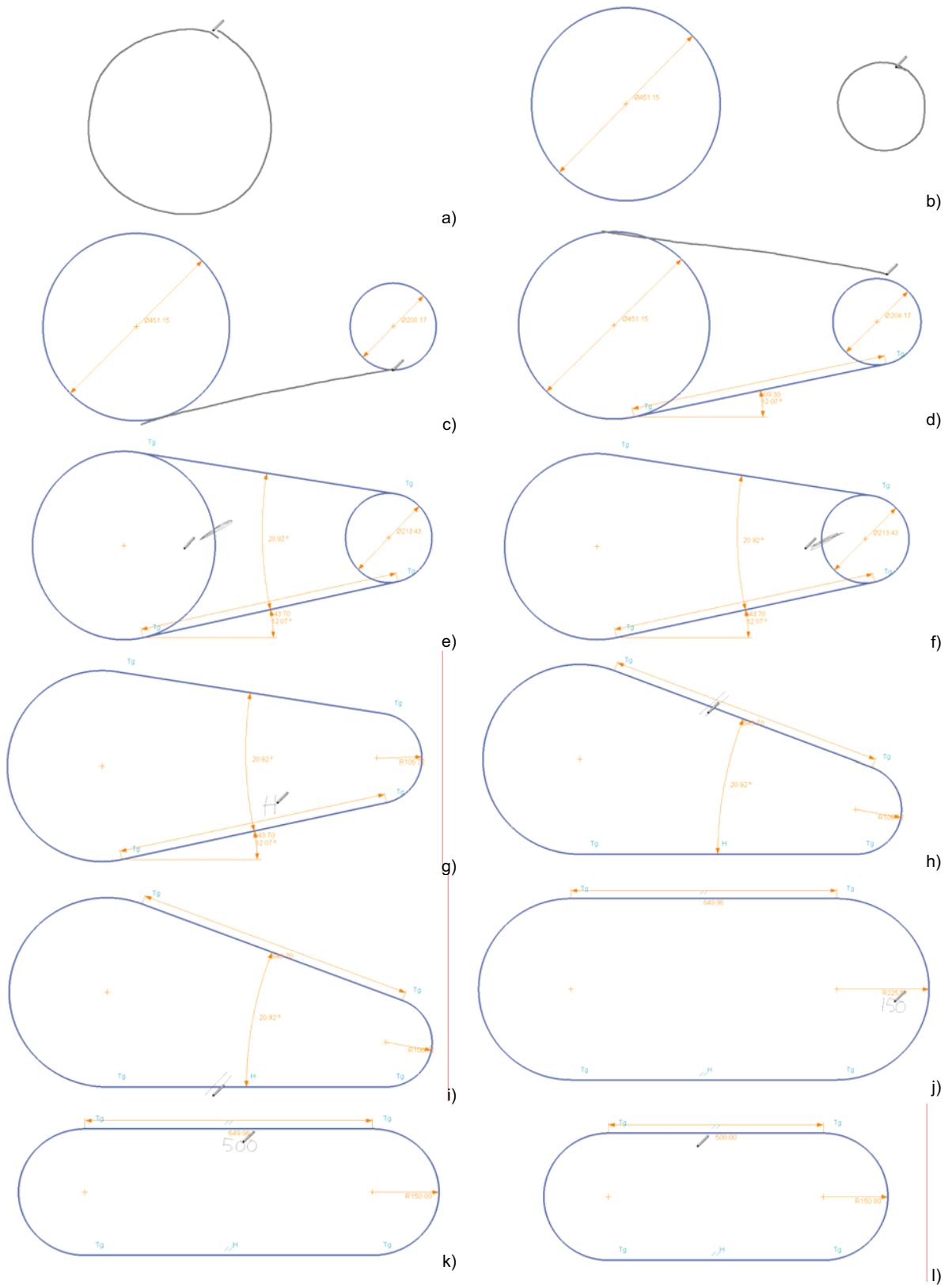


Figure 3. Drawing sequence in GEGROSS

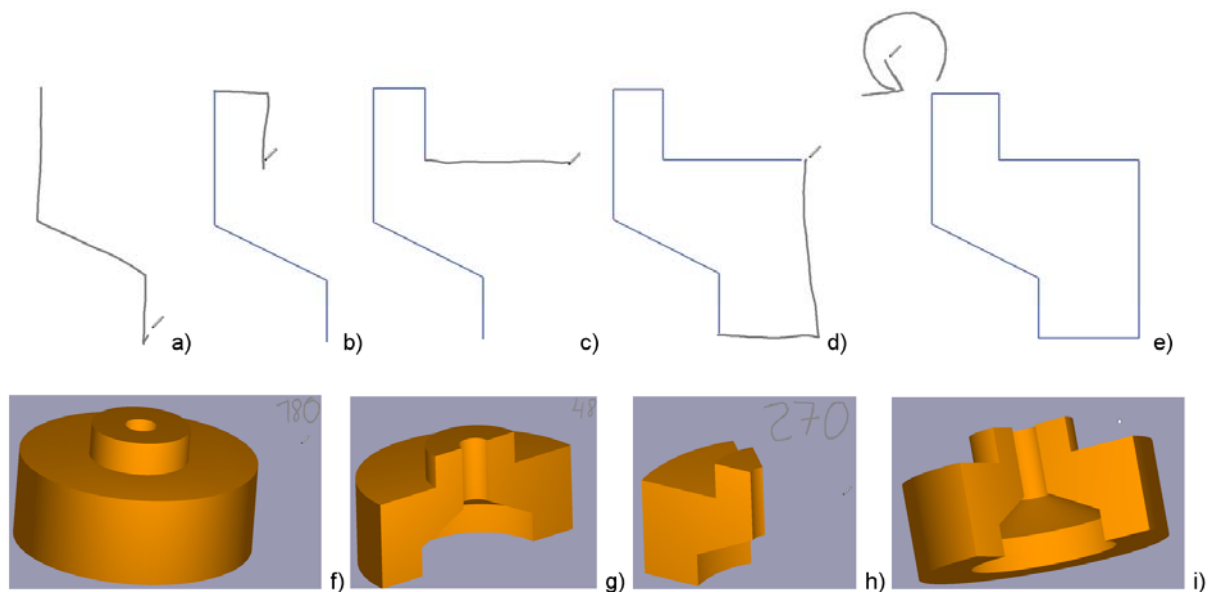


Figure 4. Modelling sequence in GEGROSS. Example of revolution shape

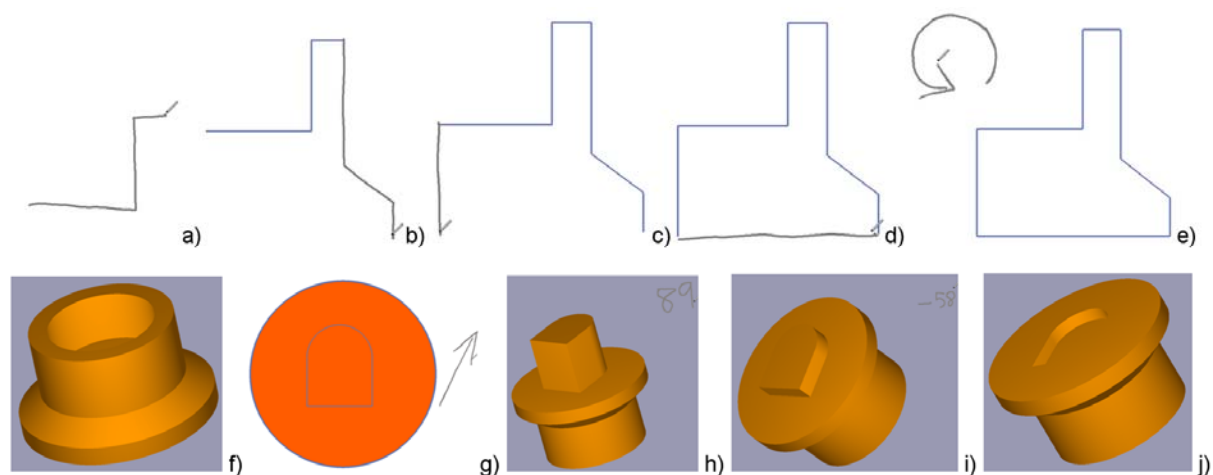


Figure 5. Modelling sequence in GEGROSS. Example of revolution/extrusion shape

The drawing beautification process works transparently to the user and cleans up input data and adjusts edges to make sure they meet precisely at common endpoints in order to get geometrically consistent figures, filtering all defects and errors of the initial sketches that are inherent to their inaccurate and incomplete nature. Moreover it is possible to search dimensional and geometric constraints. Some of the drawing aids implemented are: coincident (if a coincident constraint is defined between a point and any geometry then this implies that the point lies on the geometry), concentric, parallel, tangent, equal radius (it implies that the radii of the geometries are the same), perpendicular, equal distance (this constraint is used for search geometries with the same length), distance, angle and radius.

Snapping and adjustments are performed in real time, and the user can see the effect. The user can reconfigure the set of constraints that the system will automatically use to control the sketch. Users can also manage the beautification action by modifying the threshold values used to decide whether a geometric constraint is verified. In other words, a general tolerance parameter controls the beautification action, because some users prefer a less automatic drawing control (Figure 7). Constraints are managed by this dialog box and can

choose the constraints that the system will automatically use to control the sketch. Also the user can modify the search sequence of constraints.

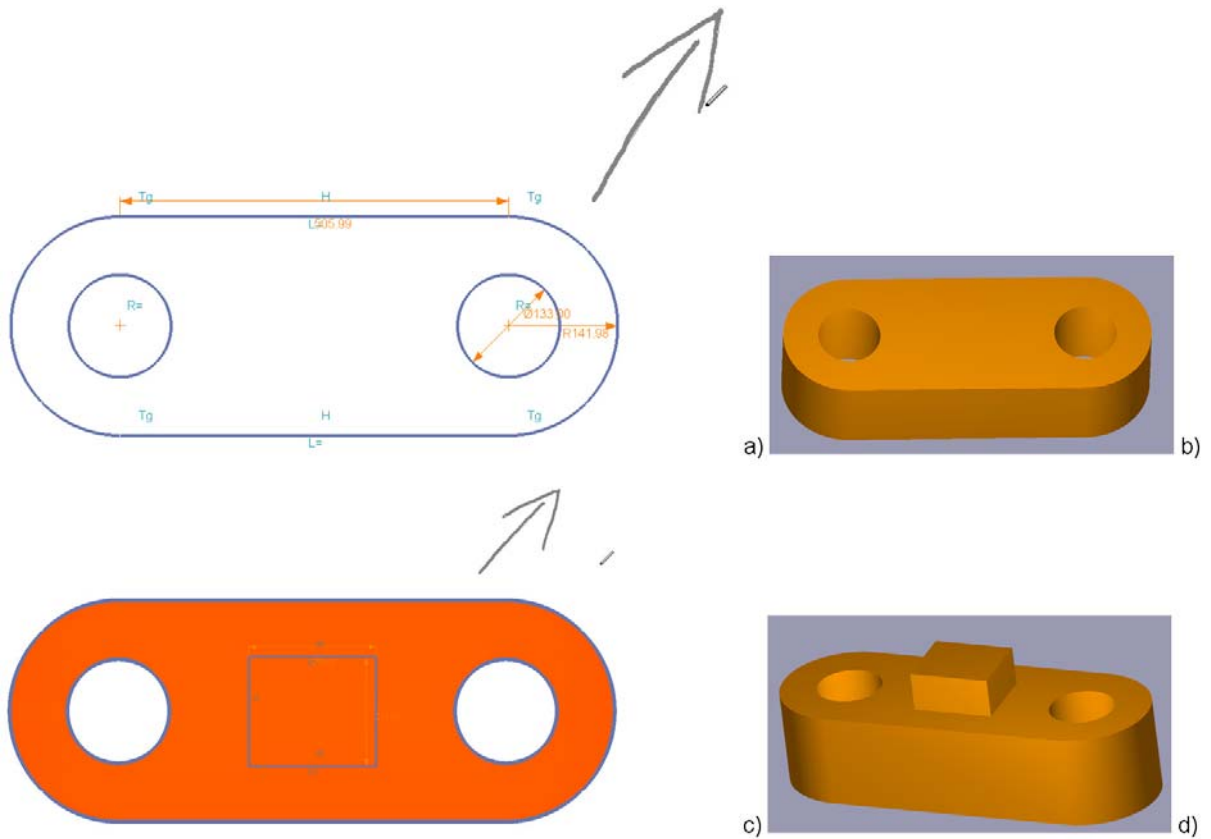


Figure 6. Modelling sequence in GEGROSS. Example of extrusion shape

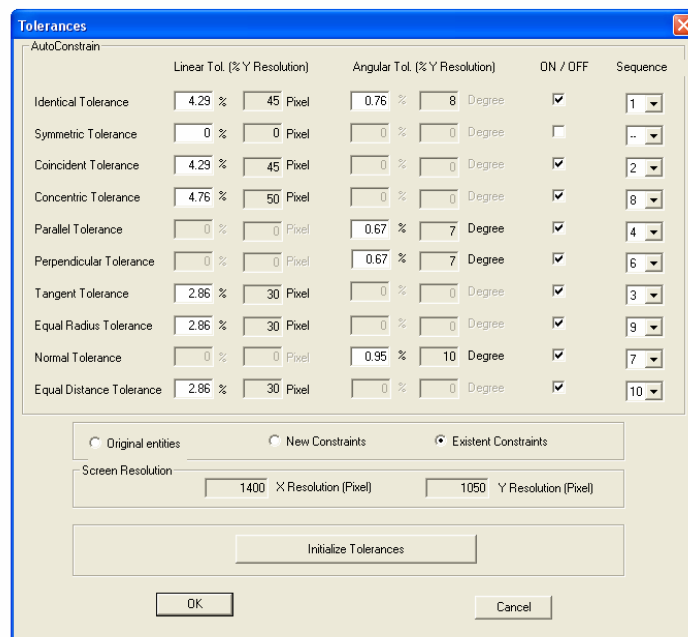


Figure 7. Tolerance settings

4. Conclusions

An approach to create three-dimensional parametric objects using a gesture alphabet has been described. The main objective of this work has been to provide dimensional and geometrical control over the sections in an easy and natural way. With this purpose, handwritten dimensions offer a natural method that is known by any engineer. Preliminary usability tests have shown encouraging results (Naya et al., 2007b). Users that have an engineering background find very natural the system behaviour, and the learning process to manage the application is very fast.

Comparing the operation of the GEGROSS system with a standard WIMP parametric CAD application we can say that the basic functionality is practically equivalent. As can be seen in previous examples, usability is enhanced as well since the interface has been particularly tailored to detect standardized symbols. New symbols are not invented for any existing ones, but they are “borrowed” from the set of meaningful engineering symbols currently defined in the standards (ISO, ASME...). Therefore, the improvement in usability results from the fact that those symbols are commonplace for potential users: no learning is required, and unconscious user actions are readily interpreted by the computer. User satisfaction has been very high during the usability study. Users enjoy the simplicity of the system and its powerful control of geometry.

References

- Cera, C.D., Regli, W.C., Braude, I., Shapirstein, Y. and Foster C.V. (2002), “A Collaborative 3D Environment for Authoring Design Semantics”, *IEEE Computer Graphics and Applications*, Vol. 22 (3), pp. 43-55.
- Computers & Graphics* (2000), Vol. 24 (6).
- Contero, M., Naya, F., Company, P., Saorín, J.L. and Conesa J. (2005), "Improving Visualization Skills in Engineering Education", *IEEE Computer Graphics and Applications*, Vol. 25 (5), pp. 24-31.
- De Araujo, B. and Jorge, J. (2003), “BlobMaker: Free-form modelling with variational implicit surfaces”, *12° Encontro Português de Computação Gráfica*, pp. 17-26, Porto.
- Fonseca, M. and Jorge, J. (2001), “Experimental Evaluation of an On-Line Scribble Recognizer”. *Pattern Recognition Letters*, Vol. 22 (12), pp. 1311-1319.
- Igarashi, T. and Zeleznik, B. (2007), “Sketch based interaction”, *IEEE Computer Graphics and Applications*, Vol. 27 (1), pp. 26–27.
- Igarashi, T., Matsuoka, S. and Tanaka, H. (1999), “Teddy: a sketching interface for 3D freeform design”, *Proceedings of ACM SIGGRAPH '99*, pp. 409-416, Los Angeles, California.
- Naya, F., Contero, M. and Aleixos, N. (2007a), "The Mobile Drawing Assistant ", *International Journal of Engineering Education*, Vol. 23 (3).
- Naya, F., Contero, M., Aleixos, N. and Company, P. (2007b), "ParSketch: A Sketch-Based Interface for a 2D Parametric Geometry Editor", *Lecture Notes in Computer Science*, Vol. 4551, pp. 115-124 .
- Pereira, J., Jorge, J., Branco, V. and Nunes, F. (2000), “Towards calligraphic interfaces: sketching 3D scenes with gestures and context icons”, *WSCG'2000 Conference*, Skala V. Ed.
- Schweikardt, E. and Gross, M.D. (2000), “Digital Clay: deriving digital models from freehand sketches”, *Automation in Construction*, Vol. 9 (1), pp. 107-115.

Turner, A., Chapmann, D. and Penn, A. (2000), "Sketching space", *Computers & Graphics*, Vol. 24 (6), pp. 869-879.

Tversky, B. (2002), "What Do Sketches Say About Thinking?", *AAAI Spring Symposium Series - Sketch Understanding*, pp. 148-152.

Zelevnik, R.C., Herndon, K.P. and Hughes, J.F. (1996), "SKETCH: an interface for sketching 3D scenes", *Proceedings of SIGGRAPH '96*, pp. 163-170.

Acknowledgements

The Spanish Ministry of Science and Education and the European Union (Project DPI2004-01373) supported this work. It was also partially supported by Fundació Caixa Castelló-Bancaixa under the Universitat Jaume I program for Research Promotion (Project P1-1B2004-02).

Correspondence (For more information please contact)

Ferran Naya Sanchis
DIG – ETSII, Polytechnic University of Valencia. Camino de Vera s/n, 46022 Valencia, Spain.
Phone: +34 96 3879517
E-mail: fernasan@dig.upv.es