

ParSketch: a Sketch-based Interface for a 2D Parametric Geometry Editor

Ferran Naya¹, Manuel Contero¹, Nuria Aleixos², and Pedro Company²

¹ Polytechnic University of Valencia, Camino de Vera s/n,
46022 Valencia, Spain

{fernasan, mcontero}@degi.upv.es

² Universitat Jaume I, Avda. Sos Baynat s/n,
12071 Castellón, Spain

{naleixos, pcompany}@emc.uji.es

Abstract. ParSketch is a software prototype to evaluate the usability and functionality of a sketching interface aimed at defining 2D parametric sections. Currently, ParSketch interprets strokes which can be recognized as geometry (line, arc, circle, ellipse, or composed entities that are automatically segmented into those basic entities), or graphic gestures representing constraints (dimension, parallel, perpendicular, tangent, concentric, horizontal or vertical). From the functionality point of view, ParSketch compares to current commercial parametric CAD applications, as it offers many of the features provided by such applications. A theoretical analysis of the efficiency component of usability is provided that justifies the potential capability of sketching interfaces to compete with classical WIMP applications. Finally, a usability study is presented, which makes special emphasis in the satisfaction component of usability.

Keywords: sketching, parametric drawing, usability of sketching interfaces.

1 Introduction

Despite the great advances in Computer-Aided Design (CAD), available commercial Graphical User Interfaces (GUIs) used by these applications have not evolved from the so-called WIMP (Window, Icon, Menu, Pointing device) interaction paradigm. These CAD tools provide a good and effective way to create the digital geometric representation of the product; however, these systems provide little support for the most creative part of the engineering design process. Usually, this part of the process, known as conceptual design, uses sketches as a powerful tool that helps designers express their creative thinking.

Majority of engineers and designers receive sketching training during their studies. A recent survey performed by the Engineering Design Graphics Division of the American Society for Engineering Education concluded that the “ability to create 3-D solid computer models” and the “ability to sketch engineering objects in the freehand mode” were the two most important graphical communication outcomes for engineering students [1,2].

If sketching is considered an important skill in engineering design; if engineers and designers usually can create sketches, as they get specific training for it, why sketching is not integrated in the digital design process? In this paper we will try to provide some reasons for the disconnection of sketching from the digital design tools, and show some of our own developments to support sketching with computers, ending with a usability study of one of these tools.

2 Sketch-based Interfaces and Modeling (SBIM)

We think that SBIM can provide the computer tools required to shift [3] to a new paradigm where sketches could be used as input to create preliminary digital engineering models in the conceptual design stage.

We are speaking about sketches, but have all the sketches the same characteristics? Have they the same purpose? Do they provide the same kind of information? We have followed the classification proposed by Ferguson [4] that distinguishes among “*thinking sketches*” used to focus and guide non verbal thinking, “*talking sketches*” employed to support discussion on the design with colleagues and “*prescriptive sketches*” applied to give instructions to the draftsman who is in charge of making the finished drawing.

Previous researches of our group REGEO (see www.regeo.uji.es) explored and developed some applications [5,6] running on Tablet-PCs that were found to be valid to replace *thinking sketches* on paper.

Currently our research group is launching a new investigation line whose objective is to use prescriptive sketches as an alternative input method for geometry modeling in CAD systems. In the Ferguson’s definition, *prescriptive sketches* are used to give instructions (see figure 1 for an example) to the draftsman who is in charge of making the drawings. We propose an update for this concept: a virtual draftsman that interprets sketches to construct 3D models instead of drawings.

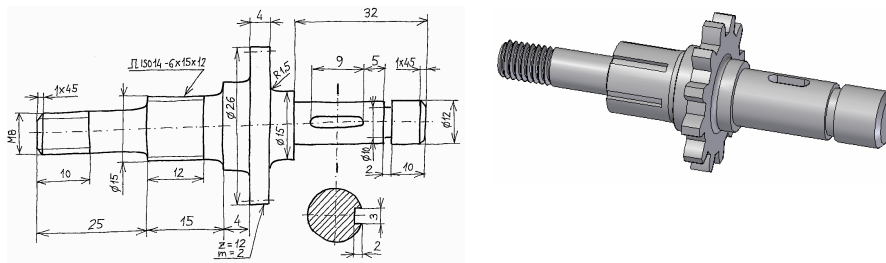


Fig. 1. Prescriptive sketch example and its associated 3D model.

A first step in this approach is to build a software prototype to evaluate the feasibility of using a sketching interface to define 2D parametric sections. The design goal of this tool is to create 2D parametric sections in a very simple way, using the conventions of technical drawing to define the shape of the section. With that aim we have developed the ParSketch application that combines a sketching interface with a commercial 2D parametric engine (UGS D-Cubed 2DCM). From the functionality point of view, ParSketch offers many of the features provided by current commercial parametric CAD applications as it is built on top of the most common parametric engine of the market.

2 The ParSketch Application: System Overview

Currently, ParSketch interprets strokes which can be recognized as geometry (line, arc, circle, ellipse, or composed entities that are automatically segmented into those basic entities), or symbols representing constraints (fig. 3 includes current supported constraints). Furthermore, drawing entities can be removed using a scratching gesture. This not only allows errors to be corrected, but also enables more complex shapes to be drawn incrementally, by refining simpler forms (see figure 2 for a sketching example). Besides, new gestures can be added to the system as the gesture recognizer can be trained by providing new gesture samples.

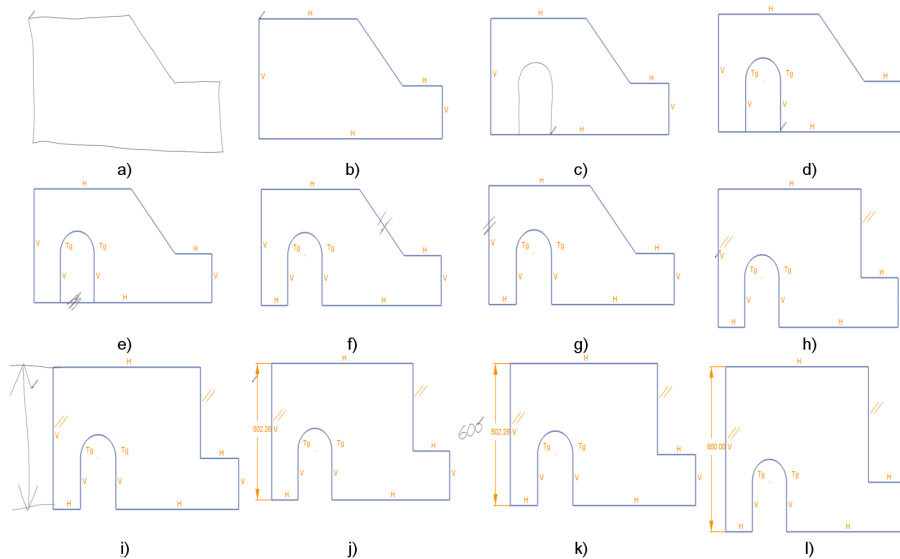






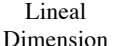
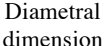
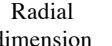


Fig. 2. Sketching sequence in ParSketch (video can be downloaded from [7]).

At present the system uses pen pressure to decide whether input corresponds to geometry or to gesture strokes. The criterion is that drawing applying high pressure means a geometry stroke; otherwise stroke is analyzed as a gesture. Both geometry and gesture analyzers make use of two geometric signatures to perform recognition: the direction and curvature graphs of each stroke,.

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

Parallel 	Perpendicular 	Tangent 
Vertical 	Horizontal 	Concentric 
Lineal Dimension 	Diametral dimension 	Radial dimension 

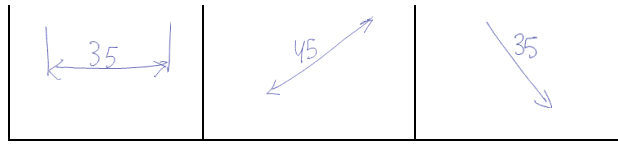


Fig. 3. Gesture alphabet.

The user introduces the geometry creating a freehand sketch directly on the Tablet-PC screen. The recognition engine cleans up input data and adjusts edges to make sure they meet precisely at common endpoints in order to obtain geometrically consistent figures by filtering all defects and errors of the initial sketches that are inherent to their inaccurate and incomplete nature.

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 (see Fig. 5 for details).

An example of interaction with the ParSketch application is provided in figure 2. In this example the user draws the whole contour in 2.a. One single stroke is accepted as input, and it is later decomposed by the application into six rectilinear and connected strokes. When the application shows the beautified version (figure 2.b), the user adds another complex stroke composed by two segments and one arc. The geometry is then beautified (figure 2.d). In figure 2.e we can see the use of the scratching gesture to refine the geometry. Drawing this gesture is interpreted by the application as an order to delete those geometric entities intersecting the smallest quadrilateral that encloses the gesture.

Then a parallel constraint is applied by simply sketching its associated gesture over the two segments we want to make parallel (see 2.f, 2.g, 2.h). Once the desired shape has been obtained, we can proceed with dimensional control. A first action is to draw a dimension without the dimension text (see figure 2.i). This is interpreted by the application as a measure command, and the current value of that dimension is shown, as seen in figure 2.j. 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 (figures 2.k and 2.l).

Handwritten number recognition is provided by the Windows XP Tablet PC Edition operative system. In this way, we provide a very natural form of imposing the desired dimensions over the sketch.

3 Sketching-based vs. WIMP interfaces for parametric drawing

From a theoretical point of view we can show that if the sketching application supports complex strokes, i.e. strokes composed by several basic primitives as line segments and arcs (see figure 4 as an example) this means a potential advantage over WIMP interaction. For instance, analyzing sections composed exclusively by arcs and line segments, we can make an approximated calculation of the number of interactions required by a WIMP application to complete the drawing task.

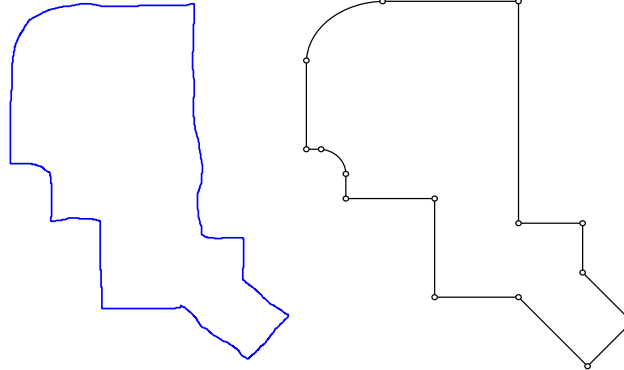


Fig. 4. Automatic segmentation vs. explicit drawing.

Usually one interaction is required to initiate the drawing process (one mouse click) and another one for finishing (a double click or pressing the enter key, for example). For drawing the line segments and tangent arcs in figure 4 two more interactions per elements are required. One is for defining the connecting vertex and the other for the selection of the proper geometric constraint as the horizontal, vertical, perpendicular or tangent conditions in this example. We count for this second interaction although, in modern parametric sketchers, geometric constraints are dynamically added as the user moves the drawing cursor. Only after the user detects the proper constraint is when he/she introduces the next entity vertex. This requires user attention, so we add it to the global number of interactions.

The last term in the calculation of total number of interactions is related to primitive switching operation. When the user wants to link for example a tangent arc to a previous polyline, he/she must spend one interaction, providing this information to the system (using for example a contextual menu or icon selection), and then spend a second interaction to come back to polyline mode.

In sum, if n_l and n_a represents the number of line segments and arcs respectively, the total number of interactions N spent by the user is:

$$N = 2 + 2(n_l + n_a) + 2n_a \quad (1)$$

Even for not too complex figures ($N= 36$ in figure 4) the last equation shows that although a user could employ several strokes to complete the shape and require some corrections to overcome recognition errors by the sketching application, there is a wide margin to compete with WIMP-based interaction in terms of efficiency. So it is feasible to implement a robust geometric segmentation and

recognition to keep advantage over WIMP interaction. We think that this is one of the keys for success in providing a real alternative or at least a complement to a WIMP interface. But as Igarashi and Zeleznik noted [8], we must adapt the design of our applications to exploit the pen input's intrinsic capacity for rapid, direct, modeless 2D expression.

To improve segmentation results, our system can be adapted to each user way of sketching by means of a tolerance control panel (see Fig. 5) that defines some key parameters for improving recognition. As explained before, the mode detection has been solved using the electronic pen pressure information, since ParSketch 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, we have used this information to discriminate among geometry or gesture.

In relation with other typical operations in a parametric 2D application, as imposing geometric constraints or performing dimensional control, the number of interactions required by both systems is similar. So we can conclude that from the efficiency point of view the sketch based approach is a viable option.

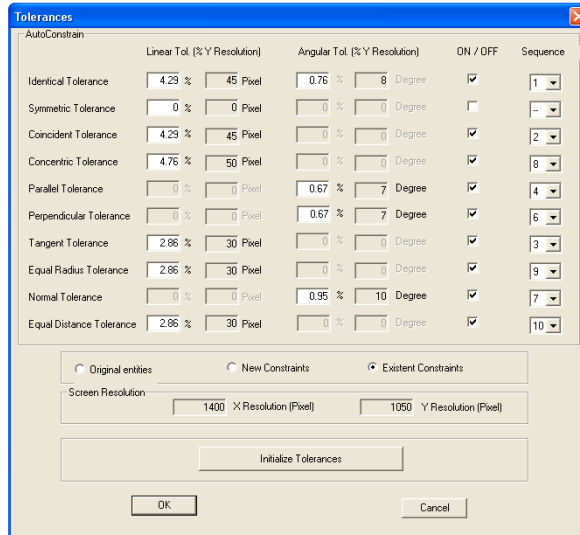


Fig. 5. Tolerance settings for controlling parametric behavior.

3.1 Usability Study

The usability of digital thinking sketches as opposed to traditional paper-and-pencil sketches was measured elsewhere [9]. In this work, we have centered our study in the user satisfaction component of usability, following the usability definition provided by ISO 9241-11, where it stands for “*extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*”. At its current development status, we are more interested in the user satisfaction response in order to decide investing more resources in the improvement of effectiveness and efficiency of the tool, although we have selected metrics for each usability component [10].

As noted previously, the main design goals of the ParSketch application are:

- Expeditious creation of shapes composed by polylines, arcs, and circles.
- Dimensional and geometric shape-control through the use of technical drawing conventions.

The evaluation involved six CAD instructors and six students with parametric CAD experience. All but one of the CAD instructors were male with an average age of 55. All students were male with an average age of 24.

We allowed 30 minutes for the evaluation, which had four parts: an overview of the system where some short videos showed the system operation, an instruction stage with a modified version of ParSketch that explicitly informs the user about the recognized entities or gestures (typically 10 minutes were employed in this training), a drawing task, and a final discussion with participants. After the discussion, users filled a questionnaire to evaluate ParSketch and express their comments about it.

Each participant used a Toshiba Tecra M4. This Tablet PC has a 14,1” screen, with a resolution of 1400x1050 dots, and employs Ms Window XP Tablet-PC Edition.

We asked users to accomplish three drawing tasks. Shapes presented in Fig. 6 were used to propose several drawing exercises. The first exercise was to create a parametric section similar to the left shape of Fig. 5. The other two exercises employed the other shapes, and the users had to create the shape and impose some dimensional and geometric constraints.

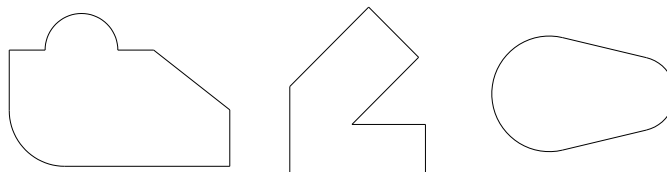


Fig. 6. Shapes for usability study.

In relation with the evaluation of effectiveness, we have measured the accuracy and completeness with which users achieved the goals presented previously, using experts' assessment of the produced sketches. This assessment is based on the accuracy of the

generated shapes. All the participants completed the requested drawing tasks satisfying all the conditions imposed to the generated shapes.

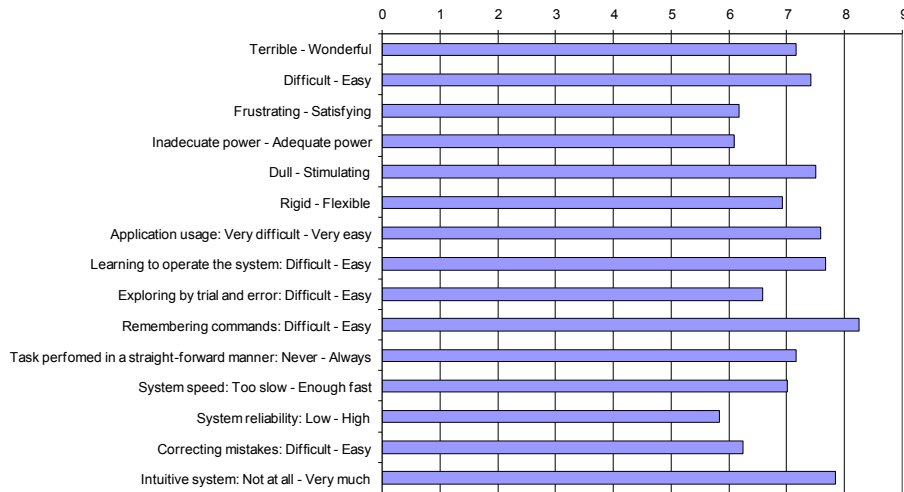
Efficiency has been measured taking into account the resources expended in relation to the accuracy and completeness with which users completed the drawing tasks. In our study we have used the task completion time and the number of events logged by a modified macro recording application. The most interesting result in this measure was the comparison between the best results obtained with ParSketch and the minimum number of interactions required by PTC's Pro/Engineer Wildfire 3 to complete the drawing tasks (this data are presented in Table 1). For all the participants in the study, this was their first contact with a Tablet-PC, and some of them had problems to control the pressure threshold that changes input mode.

Table 1. Efficiency comparison

Exercise	ParSketch (# of strokes)	Pro/E (# mouse click + # menu selection)
#1 (left)	3	12+4
#2 (middle)	1	8+1
#3 (right)	4	10+4

From Table 1 we can extract a first topic of discussion. Is it comparable the mental effort to generate a stroke on the Tablet-PC with the equivalent mouse operations to define the same geometry? We think that for users with previous experience in sketching on plain paper, drawing is practically an automatic task, which requires less concentration and effort than the mouse operation. Perhaps this justifies that 100% of participants evaluated as easier, the use of ParSketch with respect to the CAD tools known by them. Finally, user satisfaction has been measured using an adapted version of the QUIS. Questionnaire [9] using a 10 point scale from 0 to 9. A selection of the questions is presented in Table 2. In general, all participants expressed a very positive attitude towards the application, and all of them learnt in a few minutes to use it. Majority of comments about the system came from the pressure-based mode selection and about recognition errors. With respect to the pressure, none of participants had had previous experience with pressure sensible application and this had a distracting effect, requiring some concentration effort to change from the geometry input mode to the gesture one. We think that with more time of use, this mode change would not require so much effort. Also we are thinking about the convenience of providing some kind of online indicator (feedback) of what kind of input is receiving the system. Now, the application uses a paradigm similar to drawing on plain paper. Thickness of the rendered stroke in screen is related to the pressure done by the user while it is drawing. We are thinking on a color-based indication system, that will represent geometry strokes in one color, and gesture strokes in another different one. This color assignment should be done dynamically, because in this way, if the user inadvertently begins to draw the stroke in the wrong mode he/she can correct it on the fly.

Table 2. User satisfaction measures



The other part of the comments about the system came from mistakes in the recognition process. This creates some kind of frustration in the user, when he/she draws a stroke or a gesture and a wrong interpretation is provided. The recognition rate for gesture recognition was 90 percent. Rates for geometry recognition were very variable, depending on the complexity of the generated stroke and the ability of the user creating the sketch.

In order to improve recognition results, we are studying the creation of a training mode in the application to adapt and tune recognition process to each user procedure of sketching.

4 Conclusions

ParSketch offers in many cases higher *efficiency* than a comparable WIMP application. This is much related to supporting complex strokes, i.e. strokes composed by mixed basic primitives as line segments and arcs, for defining the shape's geometry.

However a high efficiency in terms of complex stroke support can have an undesired side effect: worse system effectiveness because of the increasing difficulty of the recognition and segmentation tasks. So we can conclude, than the best alternative for getting the best results is the combination of several medium complex strokes, instead of trying to define the whole geometry in one only stroke. Besides, the user can take advantage of the edition strokes (erase and the like) to follow another good strategy: recursive refinement of a first rough version of the stroke. It has some advantages. First, reduces the fail rate of the system. Second, forces the user to sketch in a more convenient way: concentrating in the major shapes, and letting the details for subsequent refinements.

User satisfaction has been very high during the usability study. Users enjoy the simplicity of the system and its powerful control of geometry. However, improvements are needed to give a clearer feedback of pressure mode selection.

Learnability of ParSketch has proven to be very high. Actually, users only have required ten minutes of introduction and demonstration before using the system. This in part is justified by the engineering background of participants. But ParSketch has been specifically designed for this kind of users, trying to exploit their knowledge of technical drawing conventions and their sketching abilities. Perhaps this is one of the reasons of this positive reaction. Users feel that this tool adapts to them, not requiring a special effort for learning to use.

Acknowledgments. 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).

References

1. Barr, R.E.: The Current Status of Graphical Communication in Engineering Education. 34th ASEE/IEEE Frontiers in Education Conference, (2004), Savannah, GA, pp. S1D8-13.
2. Barr, R.E., Krueger, T.J., Aanstoos, T.A.: Results of an EDG Student Outcomes Survey. Proceedings of the 2004 American Society For Engineering Education Annual Conference & Exposition, Salt Lake City, Utah, (2004), pp. 8-13.
3. Igarashi, T., and Zeleznik, B.: Sketch based interaction. IEEE Computer Graphics and Applications 27(1) (2007) 26–27
4. Ferguson, E.S.: Engineering and the Mind's Eye, MIT Press (1992)
5. Company, P., Contero, M., Conesa, J., Piquer, A.: An Optimisation-based Reconstruction Engine for 3D Modelling by Sketching. Computers & Graphics. 28(6) (2004) 955–979
6. Contero, M., Naya, F., Jorge, J., Conesa, J.: CIGRO: a Minimal Instruction Set Calligraphic Interface for Sketch-Based Modeling. Lecture Notes in Computer Science 2669 (2003) 549–558.
7. http://www.regeo.uji.es/videos/parsketch_01.avi
8. Landay, J. A. and Myers, B. A.: Sketching Interfaces: Toward More Human Interface Design. Computer 34(3) (2001) 56–64.
9. Company, P., Contero, M., Naya F., and Aleixos, N.: A Study of Usability of Sketching Tools Aimed at Supporting Prescriptive Sketches. Proceedings of the 4th Eurographics Workshop on Sketch-Based Interfaces and Modeling (2006) 139–146
10. Hornbæk, K.: Current Practice in Measuring Usability: Challenges to Usability Studies and Research. International Journal of Human-Computer Studies 64(2) (2006) 79–102
11. Chin, J. P., Diehl, V. A., and Norman, K. L.: Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (1988) 213–218