

**PROCEEDINGS**

# **DMS 2011**

## **The 17<sup>th</sup> International Conference on Distributed Multimedia Systems**

### **Sponsored by**

Knowledge Systems Institute Graduate School, USA

### **Co-Sponsored by**

Distributed Systems and Internet Technology Lab, University of Florence, Italy

ICSRiM, University of Leeds, United Kingdom

Visual Design and Engineering Lab, Mechanical Engineering Department,

Carnegie Mellon University, USA

Eco Controllo SpA, Italy

### **Technical Program**

August 18-20, 2011

Convitto della Calza Florence, Italy

### **Organized by**

Knowledge Systems Institute Graduate School

Copyright © 2011 by Knowledge Systems Institute Graduate School

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the publisher.

ISBN 1-891706-30-6 (Paper)

Additional Copies can be ordered from:  
Knowledge Systems Institute Graduate School  
3420 Main Street  
Skokie, IL 60076, USA  
Tel:+1-847-679-3135  
Fax:+1-847-679-3166  
Email:office@ksi.edu  
<http://www.ksi.edu>

Proceedings preparation, editing and printing are sponsored by  
Knowledge Systems Institute Graduate School

Printed by Knowledge Systems Institute Graduate School

# Sketch Input of Engineering Euclidean Solid Models

Pedro Company

Institute of New Imaging Technologies  
Universitat Jaume I  
Castellón, Spain  
pcompany@emc.uji.es

Peter Varley

Institute of New Imaging Technologies  
Universitat Jaume I  
Castellón, Spain  
varley@emc.uji.es

Raquel Plumed

Dept. Mechanical Eng. and Construction  
Universitat Jaume I  
Castellón - Spain  
plumed@emc.uji.es

**Abstract**—This position paper describes some open problems of sketch input of engineering Euclidean solid models. After a brief historical introduction, we discuss traditional design using pencil and paper, and how this paradigm has been adapted (or forgotten) by various current approaches to computer interpretation of sketches. We discuss three particular open problems, those of modalities, annotations, and assemblies. We also analyse current perception-based approaches in order to identify the most important areas in which further work is needed: detection of geometrical, perceptual and manufacturing cues, and a taxonomy of cue interdependencies which would prevent incompleteness and contradictions.

**Index Terms**—Pen computing, Sketch-based modelling, Sketching interface, Cues.

## I. INTRODUCTION

*Pen computing* refers to those computer user interfaces where users interact with computers by means of pens which guide pointers moving around a screen. In this paper, such interfaces are considered distinct from classic keyboard/display/mouse environments only as far as they are used to simulate the behaviour of actual paper and pencil. We are only interested in "virtual" paper and pencil which allows free sketching and annotating, not in the "pointing mode" of WIMP environments. Similarly, we are interested in handwriting as far as it conveys *annotations* which complement drawings, but not in handwriting recognition as a standalone problem.

Johnson et al's [1] recent overview of the evolution and current state of the art of computational support for sketching describes the advantages of sketching as part of design and the open problems which must be solved before computer-based sketching can be integrated into the design process. However, it misses some important developments in the area of Sketch-Based Modelling (SBM), which aims at developing programs capable of producing 3D geometric models from 2D sketches.

SBM aspires to encourage professional designers to discard physical pen and paper in favour of a computer-based interface in the creative phase of the design process. However, it often fails to produce the output 3D geometric model which best matches the design intent embedded in the input sketch, and its current success ratio is not good enough to persuade designers and engineers to abandon actual paper and pencil.

In order to make the problem more tractable, SBM is

generally divided into sub-problems.

The first distinction to be made is between freeform surfaces and Euclidean solid geometry (in which the object wireframe can be constructed from straight lines and arcs). Different techniques are required: the input information is different (edges for Euclidean solids vs. contour for freeform surfaces), as is the output (Euclidean solid geometry typically requires much greater geometrical exactness). In this paper, we discuss only Euclidean solid geometry—the sheer quantity of recent work on freeform surfaces (of which we highlight [2], [3], [4], [5] and [6]) makes it impractical to discuss that too.

Another important distinction is that between *perception-based* approaches, using perceptual cues to guide the interpretation process, and *geometry-based* approaches such as those of [7] and [8], which try to produce 3D models solely from the geometric information contained in the 2D input. In this paper, we argue strongly in favour of perception-based approaches: the problem of determining the embedded design intent is inherently cognitive, not purely geometrical.

The prevailing paradigm for sketch input of engineering Euclidean solid models is representative of the perception-based category of approaches [9]: (i) converting 2D sketches of wireframe drawings into 2D line drawings; (ii) deduce the faces implied by the line drawings; and (iii) use optimisation-based inflation to produce 3D geometric models from 2D line drawings. This approach was first proposed by [10], and later developed by other authors including [11] and [12]. Recent advances in face detection [13] and inflation [14] combine well, leaving vectorisation as the main open problem.

Alternative paradigms which correspond better to design engineering practice but for which the algorithms are not so well-developed include use of natural line drawings [15] or linear 3D scaffolds as guidelines for freeform shapes [16].

In choosing between paradigms, it is important to be aware how the tool conditions the task. In Section II, we discuss traditional design using pencil and paper, and how this paradigm has been adapted (or forgotten) by various current approaches to computer interpretation of sketches.

In Section III, we discuss three particular open problems, those of modalities, annotations, and assemblies.

In Section IV, we analyse the current status of optimisation-based inflation of 3D engineering models from 2D design sketches. We highlight the main limitations of this approach, and argue in favour of some strategies which may be helpful in overcoming these limitations. Improving the detection of design intent is the key to solving both problems, so detection

This work was partially funded by financial support from the Ramon y Cajal Scholarship Programme and by the "Pla de Promoció de la Investigació de la Universitat Jaume I", project P1 1B2010-01

of geometrical, perceptual and manufacturing cues embedded in the input is the first step. Then, a full taxonomy of cues and their interdependencies, aimed at preventing incompleteness or contradictions, will help to make explicit this design intent.

## II. HISTORICAL BACKGROUND

The origins of SBM can be traced back to the pioneer work of Perkins [17], [18], who analysed how people perceive drawings as representing objects, which geometrical relationships must be obeyed, and the circumstances in which geometrical relationships can be ignored.

The area now known as SBM comes from developments which were earlier known as Geometric Reconstruction. The former goal of Geometric Reconstruction was extracting information from old engineering blueprints, essentially archaeological recovery of lost know-how. However, the task proved too difficult. The main (and still unsolved) problems were those of *vectorisation* and *annotation*.

The problem of *vectorisation* (converting raw data, patterns of adjacent black and white dots, into coherent information) was too complex. The most popular academic contributions are still based on Sparse Pixel Vectorisation (SPV) algorithm [19], but Bartolo et al.'s recent work [20] is interesting as it describes the particular problem of extracting lines from scribbled drawings.

So was the problem of *annotation*. Engineering drawings convey both 3D information represented through diverse views (main orthographic views, particular views, sections, etc) and annotations (dimensions, tolerances, etc) [21].

In the short term, the problem was solved by brute force: several CAD companies offered commercial "paper-to-CAD conversion", translation of old engineering blueprints as an additional service.

It is worth noting that architecture community continues to aim for this original goal [22]. However, in the engineering community, the main goal of reconstruction changed in the 1990s. Nowadays, most systems are oriented towards conceptual design via sketch-based modelling [23].

Thus we can note how the goal has changed over time from *vectorisation* (2D + paper  $\Rightarrow$  2D + computer) via *reconstruction* (2D + paper  $\Rightarrow$  3D + computer) to full *sketch-based modelling* (mental model  $\Rightarrow$  3D + computer). For a more detailed exposition of our view of this historical background of computer interpretation of engineering drawings, see [24] and [25].

Currently, as reported in recent surveys such as [9] and [26], SBM contains a variety of sub-problems. As we shall see in the sections below, there is no general approach which solves them all. Different *critical features* (see [9] for a taxonomy) produce different bottlenecks, and levels of development are different for each critical feature.

Design intent and CAD have been linked for a long time. However, the definition of design intent is ambiguous. In 1989, design intent was associated with design constraints and the methods of manipulating design constraints during product design activities [27]. This definition continues to be used even now. However, the word *design* in the CAD community is a

synonym of *model*: modelling is representing the design intent in some way, and design intent is a basic concept of *design for change*. The model being created is flexible through changes, but the intent remains the same. Some work has been done in the SBM sector to cope with this new understanding of design intent as design-for-change (one example would be the idea that sketching a single line and then removing the central segment implicitly conveys the design intent that the remaining two segments are collinear).

However, we prefer to understand design intent as a broader concept, a mix of: *geometry*, as far as it is linked to the shape; *psychology*, as far as it is not always explicit in the sketches; and *engineering*, as far as it is linked to the function.

When geometry dominates, design intent is mainly conveyed through geometrical features. These have already been studied as *regularities* [11], [28], [29].

The psychological component appears because information not explicitly included in the input is nevertheless perceived through *perceptual cues*. Fundamentals of perceptual cues have been studied in the general perception literature, e.g. Palmer [30] and Hoffmann [31]. More specific to SBM, Perkins [18] was probably the first to realise that looking at pictures is different from looking at objects.

When function dominates, design intent is mainly conveyed through *engineering features*, where attention is given to the machining process which creates the given geometry. For example, in Figure 1, the cylindrical holes (geometry) are naturally interpreted as drilled holes (machining features), and the blends (geometry) are naturally interpreted as rounding (machining features). Both are modifications to the geometrical skeleton.

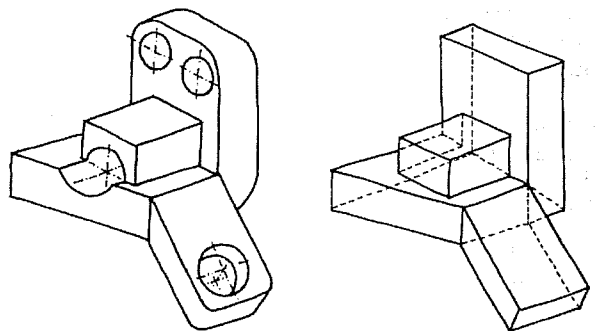


Figure 1. Geometry with features (left) and skeleton (right)

Consequently, we can define design intent as the set of intentions in sketches, conveyed through cues, which, when perceived, reveal regularities or features of the object.

It should be noted that only a few design intent cues have been studied in any detail—examples include face planarity, edge parallelism ([15]) and bilateral symmetry ([32], [33], [34] and [4]). There are clear examples where design intent should be studied further.

## III. OPEN PROBLEMS WITH VIRTUAL PAPER AND PENCIL

This section discusses three particular open problems.

### A. Pencil and Paper Modalities

How do engineers actually use pencil and paper in practice? Do current sketching tools offer all the modalities which users of pencil-and-paper expect? What is missing?

User studies such as [35] assert that current SBM tools are still less usable than pencil and paper sketches. The “hardware” of pencil and paper sketching is simple, but its operation is sophisticated as pencil and paper sketching is multimodal. Note, for example, the scaffolding lines, highlighting and hatching in Figure 2, the different inking of axes, hidden lines and autocorrection in Figure 3, the different uses of overtracing (for decoration and for “thinking over the line”), and the annotations in Figure 4.

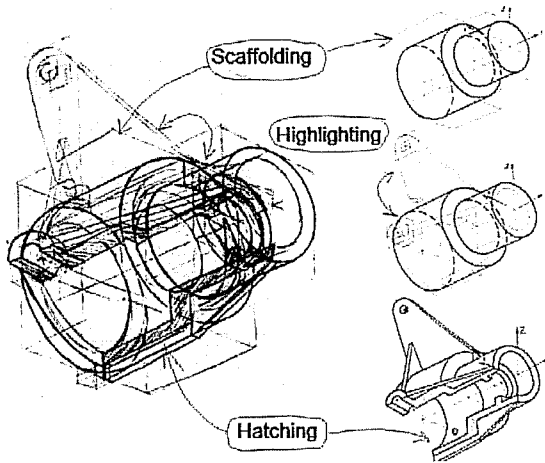


Figure 2. Multiple Modes (I)

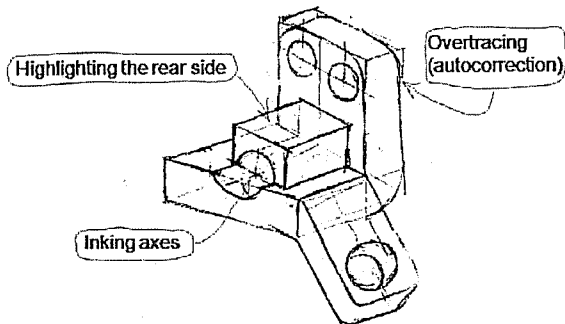


Figure 3. Multiple Modes (II)

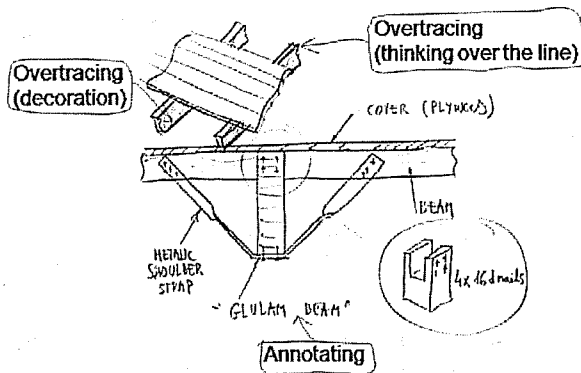


Figure 4. Multiple Modes (III)

The “overloaded” use of thin lines in Figure 3 is interesting: the human eye can easily distinguish scaffolding lines, hidden lines and axes, even though their graphic representation is the same. It is also worth noting the informal mixing of views in Figure 4: the same sketch contains an orthographic view, a detailed view and a pictorial view.

Thus we can conclude that the set of operating modes available in pencil and paper sketching is complex. Using just the pencil, we have: *views of the object* (orthographic, detailed, pictorial); *scaffolding*; *highlighting*; *hidden lines*; *axes*; *hatching*; *overtracing* (autocorrection, thinking, decoration); and *annotating*. What is more, the switching strategy is automatic and non-intrusive.

Some typical computer operations are also possible with pencil and paper sketching, albeit with a change of tool. For example, lines can be *erased* using correction fluid, a literal *cut and paste* can be performed using scissors and glue, and *regularities* can be created by tracing over a displaced copy of the original. However, in these specific cases, the computer already has an advantage.

This list of operating modes is illustrative, but far from exhaustive. We still require a full taxonomy of operating modes, including their mutual relationships and the cues used to discriminate between them.

There are two possible strategies we can take when we aim to reduce the gap between pencil and paper and SBM tools: either we try to replicate pencil and paper as closely as possible, or we try to add extra features to compensate for the loss of some of the power of pencil and paper.

Until recently, SBM tools have aimed to provide as many pencil and paper modes as possible. However, they have been based around the wrong interface paradigms, those of command-driven and menu-driven interfaces—the greatest advantage of the pencil is that it requires no commands or menus to use. But identifying an alternative interface paradigm is a difficult task in itself, and we do not know of one which would suit experienced design engineers.

In terms of adding extra features, there is broad agreement on some of the main advantages which computer-based systems already possess: it is easier to edit work; it is easier to file work; and it is easier to interface work to other applications. This list could easily be extended by adding some current CAD operations which help in reducing editing tasks: *extrusions* and *sweeps*.

But more work is needed. To achieve our final goal, making CAD tools as easy to use as pencil and paper, we need both hardware advances and software advances. Two example problems illustrate this. Firstly, current graphics tablets have proved to be less comfortable to use because of the small gaps in both time and location between the pen tip and the cursor [36]. Secondly, use and maintenance of computers still require technical knowledge which several designers have told us (and we can only agree) should not be part of their job.

User studies also assert that current SBM tools do not possess significantly improved functionality [37]. Before we can implement SBM properly, two questions must be posed: how many functions can be provided without buttons and

menus? and how many functions does a design engineer require?

Clearly, if the second answer is larger than the first, a new interface paradigm is required.

It is already clear that full modeless operation is not the goal. If we wish to replicate real pencil and paper scenarios in a virtual environment, we must be aware that real pencil and paper sketching includes a rich variety of modes. Replicating pencil and paper scenarios in virtual environments is still infeasible—although the goals of SBM have supposedly been accomplished, practical implementations have unfriendly interfaces.

So far, three types of input have been considered for SBM: perfect line drawings, line drawings containing minor geometric errors, and freehand sketches. But this too is an oversimplification, and in need of refinement. For more details on this topic, see [38].

### B. Annotated Sketches

This section discusses the problem of interpreting annotated sketches. Which strokes are annotation rather than object? What does the annotation mean? How should it be applied?

In general, we can consider three purposes for sketching: *thinking*, *talking*, and *prescribing* [39]. We can combine this with two levels of geometric information: *line drawings* and *sketches*; and with two levels of non-geometric information: *with annotations*, and *without annotations*. The particular open problem we discuss here is that of interpreting *annotated engineering sketches*.

*Annotations* is a generic term. It includes: *dimensions*; *cut views with hatching*; *icons*; *symbols*; and many other standardised conventions.

The proposed approach for producing 3D models from annotated engineering drawings is: (i) capture and record the data; (ii) separate annotation data from drawing data; (iii) interpret the drawing data (without annotations); (iv) interpret the annotations (separately from the drawing)—see, for example, [38]; (v) apply the interpreted annotations to the interpreted drawing.

The subtask of interpreting engineering symbols is not trivial. Behind even apparently quite simple drawings, there are hundreds of standards defining the exact meaning of many symbols and conventions (ISO, DIN, BSI, ANSI, ...). It is obvious that communication of relevant information depends on the meaning of symbols (see, for example, [40] for historical examples where misunderstanding of symbols has caused information loss).

The problem becomes still more challenging when we recognise that new standards (e.g. ASME Y14.41-2003) already allow annotations in 3D models.

Currently, computers are blind to these annotations—the annotations are just labels added to the model. The user can read and modify them, but the geometric engine does not use them either to construct, to edit or to validate the model.

One interesting open problem area is that of interpreting sketched data input for Computer-Aided Engineering (CAE) applications. The particular problem here is that data is input

through either of two WIMP interfaces: stand-alone CAE preprocessors which define both geometric data and attributes, and combinations of CAD applications (which export the geometry) and downstream CAE preprocessors (which add attributes).

What we want is a tool which takes as its input the sketches which designers typically draw in order to fix their ideas before interacting with CAD preprocessors, and which creates as its output a file which meets the specification of the desired analysis tool. It would be based on two reasonable assumptions: that the input sketches are drawn directly onto a computer screen which acts as virtual paper and pencil; and that the user is still in the process of conceptual design and is not ready to progress to a detailed design stage.

Hence our goals are: to supply the user with a computer interface similar to pencil and paper; to minimise the amount of information required from the user; and to give the user more freedom in inputting and editing this information.

Examples of such tools include Pre/Adef [41] and FEAsy [42]. Although they represent the current state of the art, they remain unsatisfactory. Their advantages are: they produce valid output (whereas all that pencil and paper can do is fix ideas), no training is required, and the user is not forced by the system to add unnecessary information such as dimensions. Their disadvantage is that explicit mode-switching is required (although only when switching to a very different task) and that practice has shown that users do not always feel comfortable with an on-line parser. Such interfaces are similar to, but not yet as good as, real paper and pencil.

### C. Assemblies

This section considers the possibility of sketching assemblies of parts.

Currently, we are limited to reconstructing isolated parts. However, engineering parts typically work in combination with one another. Our colleagues in mechanical design engineering sketch not only single parts but also assemblies of several parts. If they are to be persuaded to use SBM systems, SBM systems must support sketching of assemblies, perhaps by defining and implementing a set of symbols which can help an SBM system to assemble 3D models obtained from 2D sketches.

The basic guidelines of such an approach should be that: the symbols must themselves be sketched, as part of a natural design process; the meanings of the symbols must be robust (in the sense that they must be correctly interpreted by the geometric engine in charge of assembling the parts); and the symbols should not be subject to the faults of current sets of CAD operations.

We must therefore analyse what is wrong with current CAD applications. In current applications, components can be positioned within the product assembly using either reminiscent absolute coordinate placement methods, or *mating conditions*. Mating conditions are definitions of the relative positions of components with respect to one another, for example alignment of axes of two holes, or distance of two faces from one another. The final position of all components based on these relationships is calculated using a geometry constraint engine built into the CAD or visualisation package.

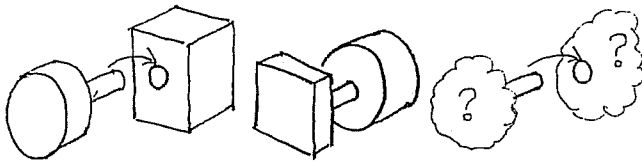


Figure 5. Assembling conceptual models

Commercial tools which include mating conditions are familiar, and assist the 3D CAD user to get an intuitive and friendly set of constraints (for example, SolidEdge's FlashFit option). As users place parts in an assembly, assembly relationships position new parts relative to parts already in the assembly. There are several possible assembly relationships.

However, we find a major drawback. Only complete and consistent parts can be assembled. Current CAD packages can assemble concrete shapes (e.g. Figure 5 left), and even modify them after assembling (e.g. to Figure 5 centre), but cannot assemble blurred or incomplete shapes (e.g. Figure 5 right). We need a complete design before assembling may proceed.

Our vision is creating a sketch-based environment in which design intent can be specified, so that we are able to assemble different parts which are not yet fully defined.

#### IV. CUES AND DESIGN INTENT

As noted above, when geometry dominates, design intent is mainly conveyed through geometrical features or *regularities*. Algorithmic solutions such as [43] can process this information (although they could still benefit from improvement).

In other cases, cues are based on perception, and we meet a problem: most researchers in perception only consider looking at physical objects, not looking at pictorial representations. Hence, although perceptual cues have been studied in the literature, applying them to produce explicit geometric constraints which may guide the construction of 3D models has barely been considered. We know of nothing in this area which goes beyond the limited detection of features in [15].

Finally, in *engineering features* attention is given to the machining process which creates the given geometry. Some cues linked to simple manufacturing features have already been studied. Extrusion is one such [44]. Rounds and chamfers are another simple but very frequent manufacturing feature [45]. Some interactions between features, such as the one shown in Figure 6 (left), interaction between two rounds, have also been studied, but there are many which have not.

When we detect several cues, the geometric criteria derived from these cues may well contain contradictions. Furthermore, as we add more cues, the mutual dependencies arising between them will lead to redundancies amongst the resulting geometric constraints, a problem which some authors (e.g. [28]) aim to detect and circumvent. General strategies to cope with these problems have already been proposed. For example, multiagent-based management of cues was first proposed by Juchmes and Leclerc [46], and [47] has recently reported that it is feasible to simultaneously detect multiple cues.

Nevertheless, a simple example is enough to show that much more work is required here. Consider the regularities which appear in the upper face of the parallelepiped

represented in Figure 6 (right): 1) a face planarity constraint on  $(a, b, c, d)$ ; 2) two parallelism constraints  $(a/c$  and  $b/d)$ ; 3) two perpendicularity constraints  $(a \perp b$  and  $a \perp d)$  detectable by line orthogonality; 4) four skewed symmetries (with axes  $e_1, e_2, e_3$  and  $e_4$ ); and 5) four orthogonal corners  $(ab, bc, cd, da)$  detectable by line orthogonality. Thus just one quadrilateral face can generate thirteen (or more) constraints. Considering the parallelepiped wireframe as a whole, there are similar sets of constraints for the remaining five faces, giving us a total of 78 constraints. Yet, if we use the classical optimisation approach to inflate the model, we have just eight variables: the z-coordinates of the vertices. With more constraints than variables, but no inconsistencies, it is clear that some constraints must be redundant.

We can note that, back in 1971, the third paper in [18] recommended finding the "geometrically maximal combination of constraints". This hides a problem which, forty years later, has still not really been solved: how geometric constraints interact in 3D.

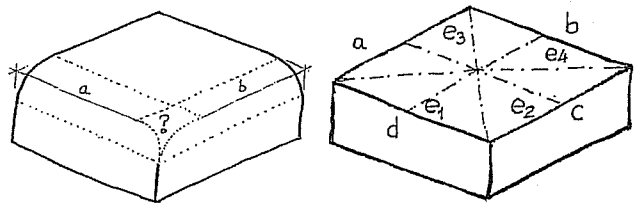


Figure 6. Interdependencies between features (left) and cues (right)

#### V. CONCLUSION

WIMP interfaces are not appropriate for conceptual design stages. However, SBM tools are not yet used—they look suitable for the task, but need improvement. We can classify problems into (a) those where a reasonably good solution exists (improvements may still be possible) and (b) open problems.

We have studied three open problems of sketching interfaces which discourage designers from using virtual paper and pencil, those of modalities, annotations, and assemblies.

We have also described the main weaknesses of cue detection and the process of finding the underlying implied design intent. We understand the capture of complex design intent from input sketches as a mix of geometry, psychology and engineering. The detection of cues needs improvement in all three areas, but is most critical in the areas of perceptual cues and engineering features.

Developing a catalogue of cues and their interdependencies should help to reduce current problems of incompleteness and contradictions which cause malfunctions in the strategies (such as optimisation algorithms) used for computer interpretation of sketches.

#### REFERENCES

- [1] G. Johnson, M. D. Gross, J. Hong, and E. Y. Do, "Computational support for sketching in design: A review". *Foundations and Trends in Human-Computer Interaction*. Vol. 2, No. 1, 1-93, 2009.
- [2] L. B. Kara, K. Shimada, "Sketch-based 3D-shape creation for industrial styling design", *IEEE Computer Graphics and Applications*, 27 (1), pp. 60-71, 2007.

- [3] A. Andre, S. Saito and M. Nakajima, "CrossSketch: Freeform surface modeling with details", 4th Eurographics Workshop on Sketch-based interfaces and modeling, Riverside, California, August 02-03, 2007.
- [4] S. Bae, R. Balakrishnan and K. Singh, "iLoveSketch: As-natural-as-possible sketching system for creating 3D curve models", ACM Symposium on User Interface Software and Technology, Monterey, CA, USA, October 19-22, 2008
- [5] A. Rivers, F. Durand, and T. Igarashi, "3D modeling with silhouettes", ACM Transactions on Graphics 29 (4), art. no. 109, 2010.
- [6] S. Roth-Koch, and E. Westkaemper, "The implementation of a sketch-based virtual product development", *Prod. Eng. Res. Devel.* 4:175-183, 2010.
- [7] K. Sugihara: "Machine interpretation of line drawing", MIT Press, 1986 (<http://home.mims.meiji.ac.jp/~sugihara/openbooks/mild.pdf>)
- [8] Cooper M.: *Line Drawing Interpretation*, Springer, 2008.
- [9] P. Company, A. Piquer, M. Contero, and F. Naya, "A survey on geometrical reconstruction as a core technology to sketch-based modeling", *Computers & Graphics.* 29 (6), 892-904, 2005.
- [10] T. Marill, "Emulating the human interpretation of line-drawings as three-dimensional objects", *Int. J. of Computer Vision;* 6(2): 147-161, 1991.
- [11] H. Lipson and M. Shpitalni, "Optimization-based reconstruction of a 3D object from a single freehand line drawing", *Computer-Aided Design* 28 (8), 651-663, 1996.
- [12] P. Company, M. Contero, J. Conesa, and A. Piquer, "An optimisation-based reconstruction engine for 3D modelling by sketching", *Computers & Graphics.* 28 (6) 955-979, 2004.
- [13] P. A. C. Varley and P. Company, "A new algorithm for finding faces in wireframes", *Computer-Aided Design* 42(4), 279-309, April 2010.
- [14] Tian C., Masry M., Lipson H. "Physical sketching: Reconstruction and analysis of 3D objects from freehand sketches", *Computer - Aided Design*, Vol. 41, pp. 147 - 158, 2009.
- [15] P. A. C. Varley, *Automatic Creation of Boundary-Representation Models from Single Line Drawings*, PhD Thesis, Cardiff University, 2003.
- [16] R. Schmidt, A. Khan, K. Singh, and G. Kurtenbach, "Analytic drawing of 3D scaffolds" *ACM Trans. Graph.*, Vol. 28, Nr. 5, 2009.
- [17] D. N. Perkins, *Cubic Corners*, Quarterly Progress Report 89, 207-214, MIT Research Laboratory of Electronics, 1968.
- [18] D. Perkins, *Cubic Corners, Oblique Views of Pictures, the Perception of Line Drawings of Simple Space Forms. Geometry and the Perception of Pictures: Three Studies.* Harvard Project Zero. Technical Report No. 5, 1971. <http://www.eric.ed.gov/PDFS/ED114328.pdf>
- [19] D. Dori. and Wenyin Liu. "Sparse pixel vectorization: an algorithm and its performance evaluation," *Pattern Analysis and Machine Intelligence*, IEEE Transactions on , vol.21, no.3, pp.202-215, Mar 1999.
- [20] A. Bartolo, K. P. Camilleri, S. G. Fabri, J. C. Borg, "Line tracking algorithm for scribbled drawings," *Communications, Control and Signal Processing*, 2008. ISCCSP 2008. 3rd International Symposium on , vol., no., pp.554-559, 12-14 March 2008.
- [21] D. Dori; K. Tombre, "From engineering drawings to 3D CAD models: are we ready now?" *Computer-Aided Design* 27, 243-254, 1995.
- [22] Y. Xuetao, P. Wonka, and A. Razdan, "Generating 3D building models from architectural drawings: A survey". *IEEE Computer Graphics and Applications*, 29 (1), 20-30, 2009.
- [23] D. Ullman, S. Wood, and D. Craig, "The importance of drawing in the mechanical design process", *Computers and Graphics* 14(2):263-274, 1990.
- [24] P. Varley and P. Company, "Automated sketching and engineering culture", in ed. B. Plimmer and T. Hammond, *Proc. VL/HCC Workshop on Sketch tools for Diagramming*, Herrsching am Ammersee, 15 Sep 2008, 83-92.
- [25] P. Company, M. Contero, P. A. C. Varley, N. Aleixos, and F. Naya, "Computer-aided sketching as a tool to promote innovation in the new product development process", *Computers in Industry.* 60 (8), pp. 592-603, 2009
- [26] L. Olsen, F.F. Samavati, M. Costa Sousa, J.A. Jorge, "Sketch-based modeling: A survey". *Computers and Graphics* 33 (1), 85-103, 2009.
- [27] F. Kimura and H. Suzuki, "A CAD system for efficient product design based on design intent". *CIRP Annals - Manufacturing Technology* , 38 (1), 149-152, 1989.
- [28] S. Yuan, L. Y. Tsui, and S. Jie, "Regularity selection for effective 3D objects reconstruction from a single line drawing", *Pattern Recognition Letters* 29 (10), 1486-1495, 2008.
- [29] M. Li, F. C. Langbein, and R. R. Martin, "Detecting design intent in approximate CAD models using symmetry", *Computer-Aided Design* 42 (3) 183-201, 2010.
- [30] S. E. Palmer, *Vision Science, Photons to Phenomenology*, Cambridge, MA: The MIT Press, 1999
- [31] D. Hoffmann; *Visual Intelligence. How we create what we see*, New York: WW Norton & Company, 2000.
- [32] S.J. Tate, G.E.M. Jared, "Recognising symmetry in solid models", *Computer-aided Design* 35 (7) 673-692, 2003
- [33] A. Piquer, R.R. Martin, P. Company, "Skewed mirror symmetry for depth estimation in 3D line-drawings", *GREC 2003, Lecture Notes in Computer Science* 3088. 138-149, 2004.
- [34] L.B. Kara, K. Shimada. "Sketch - based 3D shape creation for industrial styling design". *IEEE Computer Graphics and Applications*, vol. 27(1), 60-71, 2007.
- [35] P. Company, M. Contero, F. Naya and N. Aleixos "A study of usability of sketching tools aimed at supporting prescriptive sketches". *Eurographics Symposium Proceedings. Sketch-Based Interfaces and Modeling (SBIM06)*. pp. 139-146. 2006
- [36] R. Jarret and P. Su "Building tablet PC applications" Microsoft Press, 2002.
- [37] L.B. Kara, K. Shimada, S.D. Marmalefsky "An evaluation of user experience with a sketch-based 3D modeling system" *Computers & Graphics* 31, 580-597, 2007.
- [38] P. Company and P. A. C. Varley, "Operating modes in actual versus virtual paper-and-pencil design scenarios", *Intelligent User Interfaces (IU) Workshop on Sketch Recognition*, Sanibel Island, Florida, 8th February 2009.
- [39] R. Martin, P. Varley and H. Suzuki, "Perpendicularity as a key to interpreting line drawings of engineering objects", *Proc. Digital Engineering Workshop: 5th Japan-Korea CAD/CAM Workshop*, 115-120, 2005.
- [40] E. S. Ferguson, "Engineering and the mind's eye", MIT Press, 1992.
- [41] P. Company, N. Aleixos, F. Naya, P. A. C. Varley, M. Contero, and D. G. Fernandez-Pacheco, "A new sketch-based computer aided engineering pre-processor" *Proceedings of the Sixth International Conference on Engineering Computational Technology*. Civil comp Ltd. Paper-149, 2008.
- [42] S. Murugappan and K. Ramani, "FEAsy: A sketch - based interface integrating structural analysis in early design" *ASME Conf. Proc.* 2009, 743 (2009), DOI:10.1115/DETC2009 - 87727.
- [43] J. LaViola, "Advances in mathematical sketching: moving toward the paradigm's full potential", *IEEE Computer Graphics and Applications*, 27(1):38-48, 2007.
- [44] Y. S. Suh, "Reconstructing 3D feature-based CAD models by recognizing extrusions from a single-view drawing", *Proc. IDETC/CIE* 2007.
- [45] P. Company and P. A. C. Varley, "A method for reconstructing sketched polyhedral shapes with rounds and fillets", *10th International Symposium on Smart Graphics*, Banff, Canada, 24th-26th June 2010.
- [46] R. Juchmes, P. Leclercq, "A multi-agent system for the interpretation of architectural sketches", *Sketch-Based Interfaces and Modeling. Eurographics Symposium Proceedings. SBIM'04* pp. 53-61, 2004.
- [47] D. G. Fernández-Pacheco, J. Conesa, N. Aleixos, P. Company, and M. Contero, "An agent-based paradigm for free-hand sketch recognition", *Springer-Verlag Lecture Notes in Artificial Intelligence LNAI* 5883, pp. 345-354, 2009.