Abstract

In this work, the backgrounds and evolution of three-dimensional reconstruction of line drawings during the last thirty years is discussed. A new general taxonomy is proposed to make apparent and discuss the historical evolution of geometrical reconstruction and their challenges. The evolution of geometrical reconstruction from recovering know-how stored in engineering drawings to sketch-based modeling for helping in the first steps of conceptual design purposes, and the current challenges of geometrical reconstruction are discussed too.

1. Introduction

Description of three-dimensional objects geometry in a two-dimensional surface has been an academic discipline for more than two thousand years. The reverse problem is concerned on how to "recover" the geometrical and topological structure of a three-dimensional object by interpreting two-dimensional representations. Of course, *implicit* recovery actions are carried out by humans to "read" drawings, since ever. Yet, *explicit* formalization of this problem began to attract some attention only in the 1960's, when computers' development made possible some kind of automatic approaches. This problem, named *geometrical reconstruction* (or *line drawing reconstruction*, or simply *reconstruction*) implies the determination of geometrical and topological relations of all atomic parts of one object depicted in a drawing. It must not be mistake for *restitution* and *recognition*; two well-defined fields concerned with some kind of identification of objects, and not with a detailed description of its geometry.

Most of the known approaches are now in experimental stages, and are able to interpret (without many errors), almost all kind of polytopes. Interpretation of the most usual surface elements (like cylinders, spheres, etc.) is also considered by some of the approaches. Anyway, when complexity of objects increases, automatic processes usually give pass to different semiautomatic approaches.

In this article the result of an intensive bibliographical search on the geometrical reconstruction is summarized. Besides validating the criteria already established by other authors, a new general classification criterion is proposed, to make visible the historical evolution of geometrical reconstruction and their challenges.

One of the first attempts in geometrical reconstruction was to extract information from engineering plans or blueprints. This was an important goal since already existing designs suppose an important "know-how", which is stored in engineering drawings. This means that automatic solid-model generation from standardized drawings could have been the "bridge" to recover the information built-in in the thousands of old designs filed in drafting rooms. To do that, all information included in technical drawings should have been “read”. But the task proved to be difficult, because engineering drawings convey 3D information represented through *complex* views (main orthographic views, particular views, cuts, etc) and annotations (dimensions, tolerances, etc). In fact, multiple views based current approaches are usually limited just to consider main orthographic views without annotations. They do not even accept standardized conventions like particular views and sections.

However, the main goal of reconstruction community did change in the 1990’s. Nowadays, most of the systems are oriented towards conceptual design, via sketch-based modeling, and use only a sketch generated by the user as input data. Up to date, they avoid the most specialized conventions on general principles of representation and are limited to generate only proportional models, while they let for a latter phase the exact dimensioning.

During discussion, we shall argue that, the goal of geometrical reconstruction is far from being accomplished, since CAD systems have non-sequential (graphic) outputs, but accept only sequential (verbal) inputs. This is a direct consequence of the sequential nature of algorithmic languages used for programming tasks. On the contrary, design process, and in particular ideation process, need non-sequential thought. Yet it is important to notice that we do not claim for the physical implementation to become non-sequential; it is just the conceptual model (and the interaction front end) that must be “graphical”, in the sense of non-sequential. Consequently, one graphical language is required to improve the present communication between designers and CAD systems.

Among the challenges, we shall highlight that projective geometry laws are unable to resolve the problem, because it is well known that biunivocal correspondence among 2D images and 3D models does not exist. Hence, visual perception contributions are required. However, the visual perception is still not prone to algorithmic formulations because it has been little studied. Moreover, all of the current approaches contain a more or less balanced mix of geometric and perceptual principles, but the coupling is still not well solved.

2. Related work

First, a brief summary about sketch-modeling background is included to establish its link with geometrical reconstruction. After that, a reduced set of references is included in the next section. An extensive list of works on geometrical reconstruction is available at http://www.tec.uji.es/d/regeo/. But, only the references necessary to emphasize the main hints we find in its evolution are included in the paper.

Relative to references on geometrical reconstruction reviews, the book by Sugihara [78] is the most comprehensive reference to the early history of line drawings interpretation. Nagenda and Gujar [54] published a comment on eleven papers published between 1973 and 1984 on this topic, including a categorization tree. Wang and Grinstein [97] updated the categorization, and obtained a taxonomy of 3D objects reconstruction from line drawings in two-dimensional projection. The classification relied on different but dependent aspects.

Distinctions were made based on the nature of objects to be reconstructed; the internal representation of the generated model; the number of 2D views needed; the required premises, and the degree of interaction from the part of the user.

2.1 Sketching and modeling

Sketches have been present in computer aided design systems since their beginnings in the 1960's. Sutherland's Sketchpad [82] was the first program that allowed the user to create graphical images directly on the computer screen by means of a light pen. Johson's Sketchpad III [36] added three-dimensional modeling to Sutherland's system. Taggart, Negroponte [56] and Herot developed at the beginning of the 1970's the HUNCH system, composed by a set of FORTRAN programs which were designed to process freehand sketches drawn with a data tablet or light pen. This system was able to provide automatic latching of line endpoints, and detecting corners analyzing changes in stroke direction. Also, a first attempt to interpret sketches as representations of three-dimensional objects was done, as noted by Herot in [29].

However, designing a true interactive sketch based modeling system required the development of the corresponding on-line recognition techniques and the proper hardware devices. During the early 90's Microsoft intended to promote "Windows for Pen Computers", an operative system built over standard Windows with a set of extensions to support a user interface based on the interaction with a stylus and a LCD tablet. It was a commercial failure because of the small processing power of those "pen computers". Nowadays, the launch of the Tablet-PC at the end of 2002 has renewed the interest in sketch based applications.

The need for sketch-based geometric modelers in the environment of conceptual design can be traced back to the last decade [87][15][18][86]. At present, three main approaches to 3D modeling by sketching exist:

- *Gestural* systems provide predefined gesture alphabets that encode some geometric modeling operations; basically these systems substitute the selection of icons and menus by graphic gestures.
- *Reconstructional* systems apply geometric reconstruction techniques to build the object's geometry from a sketch.
- *Hybrid* systems combine the two previous approaches.

Chronologically reconstructional systems appeared before gestural ones. 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. Reconstructional systems took advantage of previous works in the offline line drawing recognition field.

Many of these systems apply a batch [60] or interactive beautification of the freehand sketch in the 3D geometry construction process [35][33]. Interactive beautification provides the user with immediate feedback because it operates as the user draws the sketch and it offers better integration with a calligraphic interface. Batch beautification allows some analysis to be implemented, for example symmetry detection, which are better carried out over the whole sketch [12].

2.2 Historical evolution of sketch-based systems

Viking [65], in 1992 is one of the first references in the new generation of sketch based modelers that integrates both sketching input and reconstruction. It supports a line-drawing of a polyhedral object as input where users can place geometric constraints. These constraints, together with a set of constraints derived from the line-drawing, are used to define vertex geometry in subsequent interpretations. We can consider this system as a WIMP application capable of interpreting sketches representing line drawings of objects. Modes of operation are selected by the user clicking on a set of buttons. Performance was very limited due to the available hardware.

In 1994, IDeS was presented by Branco et al. [6]. This system also used WIMP interaction combined with sketch input supporting perspective or axonometric drawings drawn without hidden lines. To build a model, the user draws it by hand, and once finished, he informs the system that it is 3D model representation. Alternatively, users have some modeling operations available as extrusion to create solid from elementary shapes, using icons representing the modeling operation. This system can be classified in the reconstruction category, working only with line drawings that represent solids with planar faces and trivalent vertices.

In 1995, during the 3rd Symposium on Solid Modeling and Applications two sketch based modeling application were presented. Quick-sketch developed by Eggli et al. [19] was a solid modeler which incorporated a sketching tool that supported lines, circles, arcs, or B-spline curves that were automatically recognized from raw strokes. Quick-sketch automatically determined relations, such as right angles, tangencies, symmetry, and parallelism, from the sketch input. Then, these relationships were used to clean up the drawing. A constraint maintenance system, based on gestural manipulation and soft constraints, was employed in this system. Several techniques for gesture-based definitions of solid objects were provided as extrusion, surface of revolution, ruled surfaces and sweep. The other system was presented by Grimstead and Martin [23] and it was able to generate a B-rep solid model from a single hidden-line removed sketch view of a 3D object. It worked on polyhedral objects with trihedral vertices and performed a 2D beautification stage as a previous step to 3D reconstruction using a line labeling algorithm.

In SIGGRAPH ’96, one of most cited sketch based modeling systems was presented. SKETCH [103] is a classical reference due to its gesture-based interface to approximate 3D polyhedral models. SKETCH used a gestural mode of input in which all operations were available directly in a 3D scene through a three-button mouse. The user sketched the salient features of any of a variety of 3D primitives and, following some simple placement rules, the corresponding 3D primitive was instatiated in the 3D scene. Its main contribution was to provide a gestural interface for many of the operations of the system.

Digital Clay presented in 1998 [71] supported polyhedral objects. Freehand isometric and perspective sketches were adjusted and transferred to a reconstruction browser that used Huffman-Clowes [11][31] algorithms to reconstruct the object's geometry; finally the data was exported in VRML format. This system combined freehand sketch input with WIMP interaction and supported direct editing of the reconstructed 3D model, being a representant of the reconstructional approach.

In SIGGRAPH ‘99 Teddy by Igarashi et al. was presented [34]. It is one of the most cited sketch-based systems. It supports freeform strokes to define the 2D silhouette of an object, that is triangulated using a constrained Delaunay triangulation. A chordal axis, which connects the middles of the internal edges of the triangulation is then built and used as a skeleton to elevate the surface. This system has inspired many other systems devoted to freeform surface definition by sketching. For example, Wang et al. [95] have developed a sketch-based mesh extrusion method that overcomes the Teddy-s spherical topology limitation.

Karpenko et al. presented in 2002 a system [39] with a very simple user interface, in the line of Igarashi's Teddy, using variational implicit surfaces for object modeling. It supported oversketching for redefining the boundary of a blob and provided simple navigation facilities to allow users to rotate and translate objects. Teddy did not support these features since the whole scene contains a single object. Blobmaker by De Araujo and Jorge [17] presents some improvements over Karpenko's system. The modeller allows the user to modify shapes using the merging and oversketching operators. In their approach the skeleton information generated during the inflation process of a blob is used to allow manipulation of blobs, avoiding the use of the mesh representation for the implementation of the operators.

In 2004, Alexe et al. [4] presented another freeform surface modeler that used Igarashi's skeleton extraction procedure [34]. They employ the skeleton to place implicit spheres, which are automatically blended in order to reconstruct a smooth surface. This system uses a double representation: the 3D shape and the skeleton, providing similar functionalities that previous systems, including new editing tools such as thickness control of the 3D shape, (copy/cut)-paste and, operation on the skeleton. Its main drawback is the difficulty in representing sharp edges.

Tai et al. [83] presented in 2004 a system based on convolution surfaces for the definition of freeform models starting from a silhouette curve. The generated shape has circular cross-section, but it can be modified using a sketched profile or shape parameters. However, its user interface follows the WIMP paradigm. Cuno et al. [16] have presented in 2005 a similar system, but it implements a gestural interface that solves ambiguity situations by means of a suggestive interface.

Stilton [85] is a sketching system oriented to architectural design. It supports sketch input in a VRML environment and uses a reconstruction process that uses the optimization approach based on genetic algorithms.

GIDeS [61] allows data input from a single-view projection or from several diedral views. In the case of object reconstruction from a single-view perspective, the systems contains a gesture alphabet to identify a basic set of modeling primitives such as: prism, pyramid, extrusion and revolution, among others. In addition the dynamic recognition of these modeling gestures provides the user a number of contextual icons which allow him to confirm his design assumptions. Qin et al. [66] also presented in 2000 an on-line sketching system based on fuzzy knowledge that followed the feature based modeling paradigm by applying recognition techniques to discover 3D features sketched by hand on an isometric drawing.

CIGRO [14] implements an interactive beautifier that adjusts the input sketch in real time and feeds a reconstruction engine operating on an axonometric projection. The system supports rectangular polyhedral objects and provides dynamic viewpoints that make it easy to implement an incremental modeling strategy. GEGROSS [55] extends CIGRO capabilities, transforming it into a hybrid system (using both gestural and reconstructional approaches) incorporating dimensional control over sketches by means of dimensions drawn by hand that control a parametric geometry engine. Another example of hybrid system is SMARTPAPER [72], that combines gestural commands with a 3D reconstruction approach based on Lipson's work [45].

To sum up, there are two main methods to capture designer intents: automatic (reconstruction-based) and interactive (gesture-based). Differences in the beautification process also exist. Those design intents that have a "local" impact (i.e. they affect the geometry only in the neighborhood of some element) are mostly carried out concurrently with sketching (interactively). Beautification that affects those design-intents that require global alterations of current geometry, however, are better done after sketching (in batch mode). Corners that do not meet are good examples of local beautifications, while symmetry requires a more global consideration [62].

3. Geometrical reconstruction taxonomy

Our own classification, which is described in detail in [63], is summarized in two tables, distinguishing between single view (Table 1) and multiple views (Table 2) approaches. Only a reduced set of references are included in the tables, since both tables are intended to emphasize the historical evolution, while clearly summarizing the main characteristics of each algorithm: the types of surfaces, the need for interaction, the internal representation of the 3D model, the main characteristics of the input 2D drawing, and whether or not the algorithms searches for all the possible solutions or stops just after finding the first valid one. In Table 1 the approaches are also classified in six different categories, from “labeling” to “regularities”. Such sort of classification is not included in multiple views (Table 2) as the differences between the approaches are considered less relevant.

There is an historical distinction between single-view and multiple-view approaches, and it is still relevant since the methodologies in both cases differ clearly. This distinction is also noticeably associated with the geometrical reconstruction change of goal. Obviously, multiple-view reconstruction was of capital importance for recovering designs stored in engineering drawings. Hence, many of those approaches use perfect line drawings as input, and include hidden edges and curved surfaces.

Table 1 goes near here

Our classification distinguishes between algorithms that accept flat surfaces and curved surfaces. That is to say, we distinguish between reconstruction of polytopes and the remainder. In general, the type of surfaces that a system is able to reconstruct is essential to verify the versatility of an application. But the reliability and efficiency to solve particular types of surfaces is an important aspect too.

First attempts of 2D line drawing interpretation were limited to *prototype objects*. Identification of shapes whose projections had been previously recorded was the objective. In other words, given an image of an object, the system identifies the object by first extracting a line drawing from the image and next searching for a prototype whose projection coincides with the line drawing. This approach was closer to recognition than to reconstruction.

A general solution was later obtained for reconstruction of *polyhedral objects*. Nevertheless, distinction between Eulerian and non-Eulerian polyhedral objects was sometimes necessary. In addition, the complexity of polyhedral objects was measured in terms of number of nodes and nodes “degrees” (the number of edges ending in a node), and it posed a limit to some reconstruction processes. This problem is still being considered [93].

Some other attempts were particularly concerned with reconstruction of *revolution objects* (like cylinders and cones) and *extruded objects*; two special cases of “sweep” geometry. Initially very important restrictions were necessary in the orientation of those objects. Finally, some improvements were done, and the orientation of curved objects was softened or even disappeared.

Nowadays, a wide domain of objects can be reconstructed. This includes manifold and non-manifold objects containing flat and cylindrical faces. However, reconstruction processes tend to become more prone to error when the objects involve curved surfaces.

Table 2 *goes near here*

Reconstruction systems can also be classified in terms of participation they require from the user. We can distinguish between automatic and guided systems. The aim is to detect who makes the critical decisions. Yet some guided systems require as much participation from the user that they could be classified as “intelligent” modeling systems, rather than reconstruction systems.

From tables 1 and 2 it can be concluded that most commonly used 3D objects representation in reconstruction problems is BRep (boundary representation). Yet, some attempts have been done to reconstruct CSG models (Constructive Solid Geometry), from 2D representations of *extruded objects*. Those approaches whose main result was labeling 2D drawings are indicated too.

Input comprises perfect line drawings, line drawings containing some “minor” mistakes, and sketches. The distinction was done to highlight the evolution from perfect line drawings to sketches that illustrates the change of goal from extracting engineering plans information to sketch-based modeling. Other more “academic” distinctions on input were not considered since there is an almost complete agreement on the best alternative.

For instance, it is generally assumed that only edges and contours are represented in the input. Consequently, we can say that only “standardized” principles of representation are used as input for reconstruction purposes. Sometimes it is said that only “pure” line drawings are considered. By “standardized” or “pure” we mean that texture, range, shadowing and other additional representation resources are not considered. It is important to notice that these other resources are currently used in object recognition.

Another general agreement is a limitation usually added on the point of view. In perspective projections the direction of projection cannot be parallel to any face, nor parallel to any pair of collinear edges. This constraint is named “general point of view convention”, and usually eliminates potential degeneration cases (in which, for instance, one face can project in one line, or two distinct edges can project on the same line).

The need to include hidden lines in the input drawing is judged another relevant criteria, since there is a clear separation among methods where the input includes all lines in the drawings (transparent models) and those methods that reconstruct from an input that just contains the visible edges (opaque models). In the transparent models approach, all lines must be drawn in the input, but generally there is no need to distinguish among visible and hidden lines. In the opaque models approach, the system generally infers the rear part after reconstructing the front part of the model.

Finally, the distinction among "solutions" was considered relevant in single-view approaches because it is obvious that the problem has many solutions, except in the simplest cases. Hence the strategy to find the "best" solution is relevant in general. In the beginning it was usual to find a set of solutions and let the user to choose the appropriate, i.e. the *completeness* of the approach was the goal. Nowadays, the tendency is to include more or less heuristic criteria to automatically choose the best option, i.e. the *easiness of use* is the goal. Hence, our classification of the approaches as "one solution" could have been refined by distinguishing between those approaches that stop when the first solution is found and those that include criteria to conduct the search to some sort of "best" solution.

4. Discussion

As it was said in the introduction, our aim is to make visible both the geometrical reconstruction evolution and their challenges. The evolution has been revisited in previous section. There, we have distinguished some different inputs, but we can notice that all of them are quite simple compared to current engineering drawings. Moreover, in single view reconstruction, two different "schools" can be observed. One is centered on labeling and linear programming approaches. And the other comes from primitive identification and is more or less regularities-based. Hence, we are going to discuss what we consider the geometrical reconstruction main challenges: the current situation and the tendencies in both the relation among reconstruction and engineering drawings, and among reconstruction and perception.

4.1 Reconstruction and engineering drawings

Many times, geometry definition and geometrical compatibility study are the "core" of design processes of mechanical parts, assemblies, and even small systems.

Designers make use of physical *prototypes* when face the most challenging problems, but they are expensive and slow. "Mind's eye" models can substitute physical prototypes when the designer has some expertise with the problem [20]. If the problem is more complex, mind's eye models can still be useful for the overall design, but formal models are needed too to complete the design.

In the so-called "design-by-drawing" method, geometrical design is carried out through the formalized body of knowledge known as descriptive geometry, where the physical prototypes and mind's eye models are advantageously substituted by engineering drawings, in order to fix the geometry that accomplishes with all the design specifications. See [5] for a delightful history on this matter.

When engineering drawings do contain only incomplete information and the signs and figures used are to be interpreted only in an approximate sense, the representation is said to be a *sketch*. While it is said to be a *plan* or "blueprint" when complete, exact and exhaustive information is represented. The distinction is important because sketches are not contractual documents while plans are. In addition sketches use to have a short life period while plans are filed and belong to the industries history. Furthermore, plans must be "self-contained" (they must require no complementary explanations) while sketches are usually complemented with verbal explanations and textual annotations.

In the design-by-drawing method, plans were massively used, while sketches were put aside. They served to synthesize initial ideas, but received little attention. The same happened when Computer Aided *Drawing* systems appeared (The so-called CADD systems, in the terminology of the end of the 1980´s and the beginning of the 1990´s). This was a quite logical consequence of the fact that freehand drawings are done with simple instruments (paper and pencil) while line drawings are done using "geometrical" instruments (i.e. instruments that guarantee the correctness of geometrical constructions). The main difference is, of course, the geometrical information contained in both kinds of drawings. In other words, it is "legal" to measure in a line drawing (if geometrical procedures are assured) to extract dimensional information; while only proportions and some other geometric characteristics (like symmetry, parallelism, and so on) can be roughly derived from a freehand drawing.

With the advent of CADD systems, and in the name of productivity, it grew rapidly the interest for some sort of automatic recovery of old designs stored in paper plans. In this context, the need was partially covered by multiple-view reconstruction approaches. However, it was soon realized that, for computers to run processes to explore engineering drawings, they ought to be able to extract not only iconic (signs based on geometry) but also symbolic (signs based on social conventions) information from engineering drawings. Besides, mismatches, errors and “complicity” present in all engineering drawings ought to be filtered by computers. In sum, it proved to be a difficult objective. And, more important, the need that did generate the goal did almost disappear in the end of the 1980’s because some CAD vendors and many independent software consultants did offer complete translation services from paper plans to CAD files.

Moreover, at the end of the 1980’s, CAD software produced a revolution in the design-by-drawing method, when virtual 3D *prototypes* began to be directly generated and manipulated by 3D CAD systems. Nowadays, geometrical modeling has completely replaced descriptive geometry. Hence, the new paradigm includes sketches to synthesize initial ideas. Next, geometric models are constructed and are used for analysis purposes. Finally, detailed drawings are automatically obtained to record and transmit the precise data needed for the production process. In this scenario, a new tool is still required to help designer in the phase of fixing ideas. Such a tool would be able to “capture” the ideas generated by the designer and automatically generate the design’s model. In other words, the solid model should be made “transparent” to the designer (it must be an *internal* model for the CAD system).

Certainly, some sketch-based modelers provide, early in the design process (during the idea generation phase), models that can be constructed quickly so that the design ideas can be tested. In sketch-based modelers’ speed is more important than geometrical accuracy. Yet, up to date, computer-based sketching capabilities continue to be limited and "unplugged" to CAD systems.

Moreover, design intents must be converted into sketched geometry before they can be "read" by those experimental sketch-based modelers. In other words, the system creates a digital prototype, but the creation task is not "linked" to any previous conceptual synthesis task.

In our opinion, for this purpose, some *language* oriented to creativity enhancement must be defined to improve the designer to CAD system communication.

The aim is not easy, because communication between designers and CAD systems is unbalanced in favor of programming needs. In today's state, CAD systems force designer to control a sequential flow, directed from specifications to detailed design.

Sequential nature of algorithmic languages is the reason, because those languages are in the back end of today's computer tools (and Graphical User Interfaces are not an exception to this rule). The need for programmers to define an implementation model of the process to be executed reinforces the sequential tendency. Because, for programmers, defining a process "conceptual" model ("what" the system can do) as close as possible to the "implementation" model ("how" it does it) is always the simplest solution. As a result, designer is continuously asked for *actions* (well-defined and sequential actions), to be done by CAD system. And this is not a good strategy when the designer is trying to fix "visions", that is, ill defined and non-sequential ideas.

Transparent commands (i.e. temporary interruption of an order to execute another "nested" order) can give the wrong impression that user can do almost everything in almost every moment. And, in fact, CADD systems (*drawing* systems) are highly "interactive", because they impose few limitations to "wanderer" users. But we must remember that the reason is that they are based on descriptive geometry and technical drawings: disciplines based on non-sequential languages. Unfortunately, this is not the case of real 3D CAD systems (*design* systems). CAD systems can create virtual three-dimensional models that, in turn, can be shown in pretty rendered (and, of course, graphic) images. But the construction of those models is strictly sequential. One single action follows every command, and the system turns back to the “neutral” state waiting for the next explicit command.

To sum up, a lot of work has been done, since gesture recognition libraries (like Cali [21]) and sketch-based systems exist, but our problem is that “non-verbal” thought cannot be expressed in a “verbal” language. Verbal is defined as synonymous of sequential. That is, verbal languages are based on variations of a set of signs along the time, never mind when the signs can be sounds or graphical forms. On the contrary, non-verbal (or “graphic”) languages are those in which transmission of information is based on the meaning of a predefined set of signs, but also in the spatial relations among all signs. That is, the resemblance, order, proportion and neighborhood relations present in every written communication (and necessary absent in oral communications). It must also be noticed that in non-sequential communication, the time needed and order followed to write and read the message does not affect the information.

The utopist objective would be a design system able to integrate all information contained in a sketch interactively during the sketch creation and refinement phases. Able to formalize the non-formalized ideas contained in the sketch. And able to analyze and evaluate the provisional model, and give the designer a feedback on the performance of the intended idea.

We can consider engineering drawings as a language used for communication and in this sense it is related to standardized conventions. However, languages are not only useful for communication; they play an inherent part in our thinking processes (it can be said that we use languages to “dialog” with ourselves), and, there, psychology and perception rules play the most important role.

4.2 Reconstruction and perception

What is “true” in a drawing depends on the pursuit of such representation. In fact, engineering graphics differ depending on their purpose or “audience”. The dependence is on the amount of information (required clarity, precision and level of detail) the receiver requires and/or can process. Three forms are usually distinguished:

1. made for personal use, and not meant to be understood by anyone but the individual who produced it.
2. intended to communicate to someone who understands technical drawings.
3. used to further clarify design ideas and to communicate those ideas to non-technical individuals.

In the evolution of geometrical reconstruction the emphasis has moved from geometry to perception. In the beginning, geometry was the center of attention. As justified above, this was partially due to the need to interpret what we have classified as drawings intended to communicate designs already finished. However, more and more the emphasis was moved to perception, in parallel to the growing interest in use drawings to further clarify design ideas and to communicate those ideas to non-technical individuals.

Certainly, perception has been less studied that geometry, but excellent references exist to cover the needs of an introductory study for the geometrical reconstruction community [58] [30]. Some highlights have to be done in order to take the maximum benefit from perception.

In our opinion, perception is a process to extract information from those stimuli that have value as signals. They are considered signals those stimuli that promote some sort of action. And they are actions because perception is an active and constructive process. In other words, as the process is constructive, it is done through stages (sequential) and levels (hierarchical). Hence, it is a *behavior*. Furthermore, this behavior departs from an innate base and is empowered by learning.

Because of the perception being a behavior, it requires *intention* and *attention*. Intention since it is a sequential and selective behavior. We do observe by way of successive eyeing, and only some specific aspects are considered while eyeing. In fact, this behavior justifies why caricatures work: because they simplify the images but keep and enlarge the pertinent aspects.

Because of the perception being empowered by learning, training is to be driven towards increase the attention. It is *deliberate* (we "pay attention") and *directed* to an objective (we "look for something"). This is why camouflage works: because when the aspect we are searching for is hidden, the entire object disappears.

In fact, optimization approach (like [50] in Table 1) is iterative (or sequential); deliberate, because we search for regularities (i.e. we put intention and attention towards those properties of the image that we believe to correspond to properties of the model), and, obviously, it is directed to an objective (i.e. we optimize a figure of merit). It is also hierarchical, since regularities are weighted in the objective function. However, recent contributions to linear programming approach (like [23], [89] and [88]) include sophisticated hierarchism algorithms that output good algorithmic solutions to "deliberation" on what is pertinent in the drawing to get the best characterization of the model.

In sum, more and more, the reconstruction approaches take into consideration the perception principles and laws, and, consequently, they are in the way to *artificial perception*.

5. Conclusions and future developments

The first "revolution" of graphical capabilities of computers in the design process was to assist drafting, and almost automate it. The second has been introducing interactive creation and manipulation of 3D virtual models to reduce (and almost eliminate) the need for descriptive geometry. In this work, we have argued that the next "revolution" will be to make engineering drawings to be a universal language for the whole computer aided design process, in order to reduce (and virtually eliminate) the need of data transfer among different phases in the process.

Geometrical reconstruction is going to play a fundamental role as a core technology in this process, since automatic solid-model generation from standardized drawings is the most efficient way to establish a fluid communication between designers and CAD systems. This is the challenge of 3D *reconstruction* of design models from engineering drawings, and perception must play a relevant role in this process.

Acknowledgments

This work was partially supported by *Fundació Caixa Castelló-Bancaixa* under the *Universitat Jaume I* program for Research Promotion (Project P1-1B2002-08, titled “From sketch to model: new user interfaces for CAD systems”) and the Spanish Ministry of Science and Education and the European Union (Project DPI2004-01373).

References

[1] Ah-Soon C, Tombre K. A step towards reconstruction of 3-D CAD models from engineering drawings. Proceedings of the 3rd Int. Conference on Document Analysis and Recognition, Montreal, Canada 14-16 August 1995. p. 331-334.

[2] Aldefeld B, Richter H. Semiautomatic three-dimensional interpretation of line drawings. Computers & Graphics 1984; 8(4): 371-380.

[3] Aldefeld B. On automatic recognition of 3D structures from 2D representations. Computer-Aided Design 1983; 15(2): 59-64.

[4] Alexe A, Gaildrat V, Barthe L. Interactive modelling from sketches using spherical implicit functions. Proceedings of the 3rd International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa, Afrigraph 2004, New York: ACM Press, 2004, p. 25-34.

[5] Booker PJ. A History of Engineering Drawing. London: Northgate Publishing Co. Ltd, 1979 (First Printed in 1963).

[6] Branco V, Costa A, Ferreira FN. Sketching 3D models with 2D interaction devices. Computer Graphics Forum 1994; 13(3): 489-502.

[7] Branco V, Costa A, Nunes F. Sketching 3D models with 2D interaction devices. Computer Graphics Forum 1994; 13(3): 489-502.

[8] Brown E, Wang PS. Three-dimensional object recovery from two-dimensional images: a new approach. In: Casasent DP, editor. Proc SPIE 2904, Intelligent Robots and Computer Vision XV: Algorithms, Techniques,Active Vision, and Materials Handling, 1996. p. 138-147.

[9] Chen Z, Perng D, Chen C, Wu C. Fast reconstruction of 3D mechanical parts from 2D orthographic views with rules. Computer Integrated Manufacturing 1992; 5(1): 2–9.

[10] Chen Z, Perng D. Automatic reconstruction of 3D solid objects from 2D orthographic views. Pattern Recognition 1988; 21(5): 439-449.

[11] Clowes MB. On seeing things. Artificial Intelligence 1971; 2(1): 79-112.

[12] Company P, Contero M, Conesa J, Piquer A. An optimisation-based reconstruction engine for 3D modeling by sketching. Computers & Graphics 2004; 28(6): 955-979.

[13] Company P, Gomis JM, Contero M: Geometrical reconstruction from single line drawings using optimization-based approaches. Proceedings of WSCG '99, Plzen, Czech Republic 8-12 February 1999. p. 361-368.

[14] Contero M, Naya F, Jorge J, Conesa J. CIGRO: a minimal instruction set calligraphic interface for sketch-based modeling. Lecture Notes in Computer Science 2003; 2669: 549-558.

[15] Cugini U. The problem of user interface in geometric modeling. Computers in Industry 1991; 17(4): 335-339.

[16] Cuno A, Esperança C, Cavalcanti PR, Farias R. 3D Free-Form Modeling with Variational Surfaces. The Journal of WSCG 2005; 13(1-3): 9-16.

[17] De Araujo B, Jorge J. BlobMaker: Free-form modelling with variational implicit surfaces. Proceedings of 12° Encontro Português de Computação Gráfica, Porto, 2003. p. 17-26.

[18] Dori D, Tombre K. From engineering drawings to 3D CAD models: are we ready now?. Computer-Aided Design 1995; 27(4): 243-254.

[19] Eggli L, Brüderlin BP, Elber G. Sketching as a solid modeler tool. Proceedings of the 3rd ACM symposium on solid modeling and applications. Salt Lake City, Utah, United States, 1995. New York: ACM Press, 1995, p. 313-322.

[20] Ferguson ES. Engineering and the Mind's Eye. Cambridge, MA: The MIT Press, 1994.

[21] Fonseca M, Jorge J. Experimental Evaluation of an On-Line Scribble Recognizer. Pattern Recognition Letters 2001; 22(12): 1311-1319.

[22] Geng W, Wang J, Zhang Y. Embedding visual cognition in 3D reconstruction from multi-view engineering drawings. Compuetr-Aided Design 2002; 34: 321-336.

[23] Grimstead IJ, Martin RR. Creating solid models from single 2D sketches. Proceedings of the 3rd Symp. on Solid Modeling Applications, Salt Lake City, Utah 17-19 May 1995. New York: ACM Press, p. 323-337.

[24] Grimstead IJ, Martin RR. Incremental line labelling for sketch input of solid models. Computer Graphics Forum 1996; 15(2): 155-166.

[25] Gu K, Tang Z, Sun J. Reconstruction of 3D objects from orthographic projections. Computer Graphics Forum 1986; 5(4): 317-324.

[26] Gujar UG, Nagendra IV. Construction of 3D solid objects from orthographic views. Computers & Graphics 1989; 13(4): 505-521.

[27] Guzmán A. Decomposition of a visual scene into three-dimensional bodies. Proceedings of AFIPS Fall Joint Computer Conf, 1968. p. 291-304.

[28] Haralick RM, Queeney D. Understanding engineering drawings. Computer Graphics and Image Processing 1982; 20: 244-258.

[29] Herot CF. Graphical input through machine recognition of sketches. ACM SIGGRAPH Computer Graphics 1976; 10(2): 97-102.

[30] Hoffmann D. Visual Intelligence. How we create what we see. New York: WW Norton & Company, 2000.

[31] Huffman DA. Impossible objects as nonsense sentences. In Bernard Meltzer and Donald Michie, editors, Machine Intelligence 1971; 6: 295-323. Edinburgh University Press.

[32] Idesawa M. A system to generate a solid figure from three views. JSME Bulletin 1973; 16: 216-225.

[33] Igarashi T, Matsuoka S, Kawachiya S, Tanaka H. Interactive beautification: a technique for rapid geometric design. Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'97), October 14–17, 1997. New York: ACM Press, 1997. p. 105-114.

[34] Igarashi T, Matsuoka S, Tanaka H. Teddy: a sketching interface for 3D freeform design. Proceedings of ACM SIGGRAPH '99, Los Angeles, California 8-13 August 1999. p. 409-416.

[35] Jenkins DL, Martin RR. Applying constraints to enforce users' intentions in free-hand 2-D sketches. Intelligent Systems Engineering 1992; 1(1): 31-49.

[36] Johnson TE. Sketchpad III, three dimensional graphical communication with a digital computer. PhD Thesis. M.I.T., Cambridge, MA USA, 1963.

[37] Kanade T. A theory of origami world. Artificial Intelligence 1980; 13(3): 279-311.

[38] Kang D J, Masry M, Lipson H. Reconstruction of a 3D Object From a Main Axis System, AAAI Fall Symposium on Making Pen-Based Interaction Intelligent and Natural, October 2004, p. 63-69.

[39] Karpenko O, Hughes JF, Raskar R. Free-form sketching with variational implicit surfaces. Computer Graphics Forum 2002; 21(3): 585-594.

[40] Kuo MH: Reconstruction of quadric surface solids from three-view engineering drawings. Computer-Aided Design 1998; 30(7): 517-527.

[41] Lafue G. Recognition of three dimensional objects from orthographic views. Computer Graphics 1976 (SIGGRAPH'76); pp.103-108.

[42] Lamb D, Bandopadhay A. Interpreting a 3D object from a rough 2D line drawing. Proceedings of Visualization '90, San Francisco, California 1990. p. 59-66.

[43] Leclerc Y, Fischler M. An optimization-based approach to the interpretation of single line drawings as 3D wire frames. Int. J. of Computer Vision 1992; 9(2): 113-136.

[44] Lee H, Han S. Reconstruction of 3D interacting solids of revolution from 2D orthographic views. Computer-Aided Design 2005. In press.

[45] Lipson H, Shpitalni M. Optimization-based reconstruction of a 3D object from a single freehand line drawing. Computer-Aided Design 1996; 28(8): 651-663.

[46] Liu SX, Hu SM, Chen YJ, Sun JG. Reconstruction of curved solids from engineering drawings. Computer-Aided Design 2001; 33: 1059-1072.

[47] Lysak DB, Devaux PM, Kasturi R. View labeling for automated interpretation of engineering drawings. Pattern Recognition 1995; 28(3): 393-407.

[48] Mackworth AK. Interpreting pictures of polyhedral scenes. Artificial Intelligence 1973; 4(2): 121-137.

[49] Malik J. Interpreting line drawings of curved objects. Int. J. of Computer Vision 1987; 1: 73-103.

[50] Marill T. Emulating the human interpretation of line-drawings as three-dimensional objects. Int. J. of Computer Vision 1991; 6(2): 147-161.

[51] Martí E, Regincós J, López-Krahe J, Villanueva JJ. Hand line drawing interpretation as three-dimensional objects. Signal Processing 1993; 32(1-2): 91-110.

[52] Masuda H, Numao M. A cell-based approach for generating solid objects from orthographic projections. Computer-Aided Design 1997; 29(3): 177-187.

[53] Meeran S, Pratt MJ. Automated feature recognition from 2D drawing. Computer-Aided Design 1993; 25(1): 7-17.

[54] Nagendra IV, Gujar UG. 3-D objects from 2-D orthographic views - A survey. Computers & Graphics 1988; 12(1): 111-114.

[55] Naya F, Contero M, Aleixos N, Jorge J. Parametric freehand sketches. Lecture Notes in Computer Science 2004; 3044: 114-120.

[56] Negroponte N, Taggart T. HUNCH- an experiment in sketch recognition," in Computer Graphics, edited by W. Giloi, Berlin, 1971.

[57] Oh BS, Kim CH. Progressive reconstruction of 3D objects from a single free-hand line drawing. Computers & Graphics 2003; 27: 581-592.

[58] Palmer SE. Vision Science. Photons to Phenomenology. Cambridge, MA: The MIT Press, 1999.

[59] Parodi P. The complexity of understanding line drawing of Origami scenes. Int. J. of Computer Vision 1996; 18(2): 139-170.

[60] Pavlidis T, Van Wyk CJ. An automatic beautifier for drawings & illustrations. Computer Graphics 1985; 19(3): 225-234.

[61] Pereira J, Jorge J, Branco V, Nunes F. Towards calligraphic interfaces: sketching 3D scenes with gestures and context icons. Proceedings of WSCG '2000, Plzen, Czech Republic 7-10 February 2000.

[62] Piquer A, Martin RR, Company C. Skewed Mirror Symmetry for Depth Estimation in 3D Line Drawings. Lecture Notes in Computer Science 2004, 3088:142-153

[63] Piquer A. Artificial perception of line drawings. PhD. Thesis. Jaume I University, Spain, 2003 (in Spanish). ISBN: 84-688-7061-7

[64] Preiss K. Constructing the solid representation from engineering projections. Computer & Graphics 1984; 8(4): 381-389.

[65] Pugh D. Designing solid objects using interactive sketch interpretation. Proceedings of the 1992 Symposium on Interactive 3D Graphics, Cambridge, Massachusetts, USA, 1992. New York, NY, USA: ACM Press, 1992. p. 117-126.

[66] Qin SF, Wright DK, Jordanov IN. From on-line sketching to 2D and 3D geometry: a system based on fuzzy knowledge. Computer-Aided Design 2000; 32(14): 851-866.

[67] Roberts LG. Machine perception of three-dimensional solids. Ph.D. thesis, MIT Dep. of Electrical Engineering, 1963.

[68] Ros L, Thomas F. Overcoming Superstrictness in Line Drawing Interpretation. IEEE Trans. on Pattern Analysis and Machine Intelligence 2002; 24(4): 456-466.

[69] Sakurai H. Solid model input through orthographic views. Computers Graphics 1983; 17(3): 243-252.

[70] Sastry DS, Sasmal N, Mukerjee A. Efficient categorization of 3D edges from 2D projections. Proceedings of the 3rd IAPR Int. Workshop on Graphics Recognition GREC '99, Jaipur, India 26-27 September 1999. p. 288-297.

[71] Schweikardt E, Gross MD. Digital Clay: deriving digital models from freehand sketches. Automation in Construction 2000;9(1):107-115

[72] Shesh A, Chen B. SMARTPAPER: An interactive and user friendly sketching system. Computer Graphics Forum 2004; 23(3): 301-310.

[73] Shimshoni I, Ponce J. Recovering the shape of polyhedra using line-drawing analysis and complex reflectance models. Proceedings of IEEE Conf. Computer Vision and Pattern Recognition CVPR '94, Seattle, 21-23 June 1994. p. 514-519.

[74] Shin BS, Shin YG. Fast 3D solid model reconstruction from orthographic views. Computer-Aided Design 1998; 30(1): 63-76.

[75] Shum SP, Lau WS, Yuen MF, Yu KM. Solid reconstruction from orthographic opaque views using incremental extrusion. Computers & Graphics 1997; 21(6): 787-800.

[76] Shum SSP, Lau WS, Yuen MMF, Yu KM. Solid reconstruction from orthographic views using 2-stage extrusion. Computer-Aided Design 2001; 33: 91-102.

[77] Soni S, Gurumoorthy B. Handling solids of revolution in volume-based construction of solid models from orthographic views. J.of Computing and Information 2003; 3(3): 250-259.

[78] Sugihara K. Machine interpretation of line drawings. Cambridge, MA: The MIT Press, 1986.

[79] Sugihara K. Mathematical structures of line drawings of polyhedrons-towards man machine communication by means of line drawings. IEEE Trans. on Pattern Analysis and Machine Intelligence 1982; 4(5): 458-469.

[80] Sugihara K. Picture language for skeletal polyhedra. Computer Graphics Image Processing 1978; 8: 382-405.

[81] Suh TJ, Kim WS, Kim CH. Two phase 3D object reconstruction from two-view drawings. IEICE Trans. on Info. & Sys. 1999; E82-D(7): 1093-1100.

[82] Sutherland IE. Sketchpad, a man-machine graphical communication system. Ph.D. thesis, M.I.T., Cambridge, MA USA, 1963

[83] Tai CL, Zhang H, Fong JCK. Prototype Modeling from Sketched Silhouettes based on Convolution Surfaces. Computer Graphics Forum 2004; 23(1): 71-83.

[84] Tanaka M, Iwama K, Hosada A, Watanabe T. Descomposition of a 2D assembly drawing into 3D part drawings. Computer-Aided Design 1998; 30(1): 37-46.

[85] Turner A, Chapmann D, Penn A. Sketching space. Computers & Graphics 2000; 24(6): 869-879.

[86] Ullman DG, Wood S, Craig D. The Importance of Drawing in the Mechanical Design Process. Computers & Graphics 1990; 14(2): 263-274.

[87] Ullman DG. Toward the ideal mechanical engineering design support system. Research in Engineering Design 2002; 13(2): 55-64.

[88] Varley PAC, Martin RR, Suzuki H. Making the most of using depth reasoning to label line drawings of engineering objects. Proceedings of 9th ACM Symposium on Solid Modeing and Applications ‘04, 2004. p. 191-202.

[89] Varley PAC, Martin RR. A system for constructing boundary representation solid models from a two-dimensional sketch – frontal geometry and sketch categorisation. Proceedings of the 1st Korea-UK Joint Workshop on Geometric Modeling and Computer Graphics, Seoul, Korea 6-7 April 2000. p. 113-128.

[90] Varley PAC, Martin RR. A system for constructing boundary representation solid models from a two-dimensional sketch – topology of hidden parts. Proceedings of the 1st Korea-UK Joint Workshop on Geometric Modeling and Computer Graphics, Seoul, Korea 6-7 April 2000. p. 129-144.

[91] Varley PAC, Martin RR. A system for constructing boundary representation solid models from a two-dimensional sketch – geometric finishing. Proceedings of the 1st Korea-UK Joint Workshop on Geometric Modeling and Computer Graphics, Seoul, Korea 6-7 April 2000. p. 145-158.

[92] Varley PAC, Martin RR. The junction catalogue for labelling line drawings of polyhedra with tetrahedral vertices. Interntional Journal of Shape Modeling 2001; 7(1): 23-44.

[93] Varley PAC, Suzuki H, Martin RR. Interpreting line drawing of objects with k-vertices. Proceedings of Geometric Modeling and Processing '04, 2004. p. 249-358.

[94] Waltz DL. Understanding line drawings of scenes with shadows. In PH Winston (ed.), The Psychology of Computer Vision. New York: McGraw-Hill, 1975. p. 19-91.

[95] Wang CCL, Yuen MMF. Freeform extrusion by sketched input. Computers & Graphics 2003; 27(2): 255-263.

[96] Wang W, Grinstein G. A polyhedral object's CSG-Rep reconstruction from a single 2D line drawing. Proceedings of SPIE Intelligence Robots and Computer Vision III: Algorithms and Techniques 1192, 1989. p. 230-238.

[97] Wang W, Grinstein G. A survey of 3D solid reconstruction from 2D projection line drawings. Computer Graphics Forum 1993; 12(2): 137-158.

[98] Wang W. On the automatic reconstruction of a 3D object's constructive solid geometry representation from its 2D projection line drawing. Ph.D. thesis, University of Massachusetts, Lowell, 1992.

[99] Wei X. Computer vision method for 3D quantitative reconstruction from a single line drawing. PhD Thesis, Departament of Mathematics, Beijing University, China (writen in Chinese, for an English version, see Wang and Grinstein in 1993), 1987.

[100] Wesley MA, Markowsky G. Fleshing out projections. IBM J. of Research and Development 1981; 25(6): 934-954.

[101] Yan QW, Chen CL, Tang Z. Efficient algoritm for the reconstruction of 3D objects from orthographic projections. Computer-Aided Design 1994; 26(9): 699-717.

[102] You CF, Yang SS. Reconstruction of curvilinear manifold objects from orthographic views. Computers & Graphics 1996; 20(2): 275-293.

[103] Zeleznik RC, Herndon KP, Hughes JF. SKETCH: an interface for sketching 3D scenes. Proceedings of SIGGRAPH '96, 1996. p. 163-170.

[104] Zhang A, Xue Y, Sun X, et al. Reconstruction of 3D curvilinear wireframe model from 2D orthographic views. Lecture Notes in Computer Science 2004; 3037: 404-411.

Table 1: Single-view reconstruction approaches

			Approach						Surfa-ce		Inter-action		Represen-tation			Input			2D hidden lines		Solu-tions	
Year	Reference	Authors	Labelling	Gradient space	Linear programming	Progressive	Primitive identification	Regularities	Planar	Curve	Yes	No	B-rep	CSG	Labelling	Perfect line-drawing	Imperfect line-drawing	Sketch	Yes	No	One	Multiple
1963	[67]	Roberts					*		*			*	*				*			*	*	
1968	[27]	Guzman	*						*			*	*			*				*	*	
1971	[31]	Huffman	*						*			*			*	*				*	*	
	[11]	Clowes	*						*			*	*		*	*				*	*	
1973	[48]	Mackworth		*					*			*	*			*				*	*	
1975	[94]	Waltz	*										*									
1978	[80]	Sugihara	*						*			*	*			*			*		*	
1980	[37]	Kanade	*						*			*	*			*				*	*	
1982	[79]	Sugihara			*				*		*		*				*			*	*	
1986	[78]	Sugihara			*				*		*		*				*			*	*	
1987	[49]	Malik	*						*	*		*			*	*				*		*
	[99]	Wei		*					*			*	*				*			*	*	
1989	[96]	Wang and Grinstein					*		*	*		*		*		*				*	*	
1990	[42]	Lamb and Bandopahay				*			*		*		*					*		*	*	
1991	[50]	Marill						*	*			*	*			*			*		*	
1992	[98]	Wang					*		*	*		*		*			*			*	*	
	[43]	Leclerc and Fischler						*	*			*	*			*			*		*	
1993	[97]	Wang and Grinstein					*		*	*		*		*		*				*	*	
	[51]	Marti et al.	*						*			*			*		*		*		*	
1994	[7]	Branco et al				*			*		*			*				*		*	*	
	[73]	Shimshoni and Ponce			*				*		*		*				*			*	*	
1995	[23]	Grimstead and Martin			*				*			*	*					*		*		*
1996	[24]	Grimstead and Martin			*				*			*	*					*		*		*
	[45]	Lipson and Shpitalni						*	*	*		*	*					*	*		*	
	[59]	Parodi							*			*			*	*				*	*	
	[8]	Brown and Wang						*	*			*	*			*				*	*	
1999	[13]	Company et al						*	*			*	*				*		*		*	
2000	[89][90][91]	Varley and Martin			*				*			*	*					*		*	*	
2001	[92]	Varley and Martin	*						*			*			*			*		*	*	
2002	[68]	Ros and Thomas			*				*			*	*				*			*	*	
2003	[57]	Oh and Kim				*			*	*	*		*					*	*		*	
	[88]	Varley et al	*		*				*			*	*			*				*	*	
2004	[38]	Kang et al.						*	*			*	*					*	*		*	
	[12]	Company et al						*	*			*	*					*	*		*	

Table 2: Multiple-view reconstruction approaches

Year	Reference	Authors	Surface		Inter-action		Represen-tation		Input		2D Hidden lines	
			Planar	Curve	Yes	No	B-rep	CSG	Perfect line-drawing	Imperfect line-drawing	Yes	No
1973	[32]	Idesawa	*			*	*		*		*	
1976	[41]	Lafue	*		*		*			*	*	
1981	[100]	Wesley & Markowsky	*			*	*		*		*	
1982	[28]	Haralick & Queeney	*			*	*		*		*	
1983	[69]	Sakurai	*	*	*		*				*	
	[3]	Aldefeld	*	*	*			*	*		*	
1984	[64]	Preiss	*	*		*	*		*		*	
	[2]	Aldefeld & Richter	*	*	*			*	*		*	
1986	[25]	Gu et al	*	*		*	*		*		*	
1988	[10]	Chen & Perng	*		*			*	*		*	
1989	[26]	Gujar & Nagendra	*			*	*		*		*	
1992	[9]	Chen et al	*			*	*		*		*	
1993	[53]	Meeran & Pratt	*	*		*		*	*		*	
1994	[101]	Yan et al	*			*	*		*		*	
1995	[1]	Ah-Soon & Tombre	*	*		*	*		*		*	
	[47]	Lysak et al	*	*		*	*			*	*	
1996	[102]	You & Yang	*	*		*	*			*	*	
1997	[52]	Masuda & Numao	*	*			*			*	*	
	[75]	Shum et al	*	*	*			*	*		*	
1998	[40]	Kuo	*	*		*	*			*	*	
	[74]	Shin & Shin	*	*	*		*		*		*	
	[84]	Tanaka et al	*	*		*	*			*	*	
1999	[81]	Suh et al	*			*	*		*		*	
	[70]	Sastry et al	*		*		*		*		*	
2001	[46]	Liu et al.	*	*	*		*		*		*	
	[76]	Shum et al.	*	*		*		*	*		*	
2002	[22]	Geng et al.	*	*	*		*	*	*		*	
2003	[77]	Soni & Gurumoorthy	*	*		*		*	*		*	
2004	[104]	Zhang et al.	*	*		*	*		*		*	
2005	[44]	Lee & Han	*	*		*		*	*		*	