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.uki.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. Naganda 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 reconstructational 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 utopian 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 hierachism 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


<table>
<thead>
<tr>
<th>Year</th>
<th>Reference</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004</td>
<td>(12)</td>
<td>Valley et al.</td>
</tr>
<tr>
<td>2003</td>
<td>(11)</td>
<td>Valley et al.</td>
</tr>
<tr>
<td>2002</td>
<td>(10)</td>
<td>Valley et al.</td>
</tr>
<tr>
<td>2001</td>
<td>(9)</td>
<td>Valley and Martin</td>
</tr>
<tr>
<td>2000</td>
<td>(8)</td>
<td>Valley and Martin</td>
</tr>
<tr>
<td>1999</td>
<td>(7)</td>
<td>Valley and Martin</td>
</tr>
<tr>
<td>1998</td>
<td>(6)</td>
<td>Valley and Martin</td>
</tr>
<tr>
<td>1997</td>
<td>(5)</td>
<td>Valley and Martin</td>
</tr>
<tr>
<td>1996</td>
<td>(4)</td>
<td>Valley and Martin</td>
</tr>
<tr>
<td>1995</td>
<td>(3)</td>
<td>Valley and Martin</td>
</tr>
<tr>
<td>1994</td>
<td>(2)</td>
<td>Valley and Martin</td>
</tr>
<tr>
<td>1993</td>
<td>(1)</td>
<td>Valley and Martin</td>
</tr>
</tbody>
</table>

**Table 1: Single-view reconstruction approaches**

- **Approach**
  - Labeling
  - Gradient space
  - Linear
  - Progressive
  - Primitive identification
  - Regularities

- **Surface reconstruction**
  - Planar
  - Curve

- **Interaction**
  - Yes
  - No

- **Representation**
  - B-rep
  - CSG
  - Labeling

- **Input**
  - Perfect line-drawing
  - Imperfect line-drawing

- **2D Solution**
  - Yes
  - No

- **Solution**
  - One
  - Multiple
<table>
<thead>
<tr>
<th>Year</th>
<th>Reference</th>
<th>Authors</th>
<th>Interaction</th>
<th>Representation</th>
<th>Input</th>
<th>2D Hidden lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>1973</td>
<td>[32]</td>
<td>Idesawa</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1976</td>
<td>[41]</td>
<td>Laffue</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1982</td>
<td>[28]</td>
<td>Haralick &amp; Queeney</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1983</td>
<td>[69]</td>
<td>Sakurai</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1984</td>
<td>[64]</td>
<td>Preiss</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1986</td>
<td>[10]</td>
<td>Chen &amp; Peng</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1989</td>
<td>[26]</td>
<td>Gujar &amp; Nagendra</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1992</td>
<td>[9]</td>
<td>Chen et al</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1993</td>
<td>[53]</td>
<td>Meeran &amp; Pratt</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1994</td>
<td>[101]</td>
<td>Yan et al</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1995</td>
<td>[1]</td>
<td>Ah-Soon &amp; Tombre</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1996</td>
<td>[47]</td>
<td>Lysak et al</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1997</td>
<td>[52]</td>
<td>Masuda &amp; Numao</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1998</td>
<td>[75]</td>
<td>Shum et al</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1998</td>
<td>[40]</td>
<td>Kuo</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1999</td>
<td>[74]</td>
<td>Shin &amp; Shin</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1999</td>
<td>[84]</td>
<td>Tanaka et al</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1999</td>
<td>[61]</td>
<td>Suh et al</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001</td>
<td>[70]</td>
<td>Sastry et al</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001</td>
<td>[46]</td>
<td>Liu et al</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td>[76]</td>
<td>Shum et al</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td>[22]</td>
<td>Geng et al</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2003</td>
<td>[77]</td>
<td>Soni &amp; Gurumoothly</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td>[104]</td>
<td>Zhang et al</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2005</td>
<td>[44]</td>
<td>Lee &amp; Han</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>