BUTTERWORTH
HEINEMANN

Computer-Aided Design, Vol. 27, No. 4, pp. 243–254, 1995

0010–4485/95 $10.00 + 0.00

From engineering drawings to 3D CAD models: are we ready now?

Dov Dori and Karl Tombre*

Conversion of engineering drawings into CAD format has so far concentrated almost solely on low-level operations, such as vectorization, basic layer separation and very limited symbol recognition. The paper discusses the reasons for the lack of serious attempts at moving into higher levels and what can be done to improve this situation. On the basis of the authors' expertise in syntactic (dimension sets) and semantic (functionalities) analysis of mechanical engineering drawings, a scheme is proposed for achieving high-level conversion of technical documents of this type into 3D CAD models.

Keywords: engineering drawings, CAD models, mechanical engineering

An engineering drawing is a graphic product definition. Prior to the introduction of CAD/CAM systems, an engineering paper drawing model was the major means of design; numerous mechanical engineering drawings of operational products still exist only in this form, and are needed for maintenance of existing systems and/or as a basis for designing the next generation of the product the model of which they describe. Paper drawings resulting from subcontractors' CAD systems which are not used by the organization that ordered them are also very common. If a paper drawing is to be converted into CAD, a total rework of the existing design must be carried out. Although commercial products in this field alleviate the burden of low level processing, such as vectorization, this is still a time-consuming, error-prone and unproductive process, which nevertheless requires skilled personnel. Since this interactive, human-intensive labour requires costly trained professionals, it is frequently avoided. This in effect amounts to ignoring the many valuable design man years put on paper of already 'debugged' products. As pointed out in Reference 1, in many cases, stores of paper drawings are still growing faster than those of CAD model files, and the conversion problem will be topical for at least the next decade. Moreover, most experts agree that the 'paperless society' dream of the 1980s is becoming more remote with each newly developed type of printer and plotter and with each decline in their prices. Humans find paper just too convenient to give it up for a 'soft' screen image which cannot be easily folded and carried. This reality is equally true for textual and technical documents.

However, the significant recent progress made in scanning and high-volume secondary storage technology has made electronic archiving of paper drawings an economically viable option. This new possibility underscores the absence of an automatic ability to intelligently process these electronic drawings so that they can be incorporated into a CAD database.

Drawings in both paper and electronic (i.e. raster images of paper drawings) media are contrasted with CAD representations, in that the latter attach semantic meaning to graphic entities, enabling them to be manipulated, to be redesigned, and to serve as a basis for computer aided manufacturing (CAM). Non-CAD-based drawings have no such inherent semantics; to be converted into CAD, they must be correctly understood and interpreted, either by a human or by a machine. In addition, complete CAD systems know about and are able to handle 3D objects and not only 2D views; true CAD conversion from paper should therefore also include the reconstruction of 3D models from 2D views.

In this paper, we try to explain why conversion techniques have been stuck for so long in low-level coding (essentially vectorization, basic layer separation and very limited symbol recognition) and what can be done to improve this situation. On the basis of our own expertise in syntactic and semantic analysis of technical drawings, we propose a scheme for achieving true conversion of engineering drawings into 3D CAD models with semantic attributes. Although we focus in this work on mechanical engineering drawings, it must be noted that other visually oriented technical documents, such as cadastral maps, plant layouts, facilities drawings, utilities drawings, flowcharts, and electric and electronic diagrams, also have distinct syntaxes and semantics that differ from one domain to another. The degree of domain specialization increases along with the height of the level of document understanding[2].

Faculty of Industrial Engineering and Management, Technion–Israel Institute of Technology, Haifa 32000, Israel
*INRIA Lorraine & CRIN/CNRS, 615 rue du Jardin Botanique, BP 101, 54602 Villers-lès-Nancy, Cedex, France
Paper received: 6 July 1993. Revised: 7 December 1993

DUAL NATURE OF ENGINEERING DRAWINGS

An engineering drawing poses a special challenge because of its dual nature. It is a mixture of two types of representations: the first is the actual object (part or assembly) described by its orthographic projections, and the second, superimposed on the first, is the annotation: a formal, standard-based language that expresses dimensioning, tolerancing and manufacturing instructions and specifications (surface quality, welding etc.). This mixture of image and formal language is the major reason for the infeasibility of adapting some general image understanding system techniques to carry out this task and for our quest for a specialized approach to solve this problem.

Because of the dual image/language nature of engineering drawings, the task of their automatic understanding is somewhat analogous to two different computer-based tasks. One is scene analysis, a well known problem in machine vision, which, in this context, is aimed at extracting the 3D shape of the object described in the drawing. The other is that of compiling a computer program written in some high-level declarative language. In spite of the apparent differences between the two, which we discuss below, inspired by the analogy between compilation and annotation understanding on one hand, and between human vision and scene analysis on the other hand, we divide the understanding process roughly into three phases: lexical (early vision), syntactic (intermediate vision) and semantic (high-level vision).

Having failed to take into account this dual nature, many document analysis systems have concentrated only on a vision-based approach: the lexical phase has been limited to extraction of basic features such as vectors, the syntactic phase has grouped these features into higher-level entities to achieve some kind of 2D pattern recognition, and the semantic phase has been reduced to looking for 'contextual' attributes to give to these entities. Typically, a 1985 feasibility study of the conversion from paper to CAD[3] only mentioned that there was 'a lot of work' to do on structural analysis techniques to achieve such high-level conversion. Because of the dual nature of drawings, however, the results produced by these systems have been too limited to be really useful for any CAD system, as they lack the needed semantics without which no 3D (or even 2D) model can be maintained.

Understandably, the operations that an engineering line drawing of any discipline needs to undergo at the early vision level are very similar. Scanning, noise removal, enhancing, thresholding, and other preprocessing operations are not significantly affected by the information content of the drawing.

Having accomplished these preprocessing tasks, we have an 'optimal' binary image stored as a raster file in some format. The next step is to recognize several basic primitives, which most technical document types have in common. These include primarily wires (a generic name for bars (straight line segments)) and circular arcs, accompanying text and domain-specific symbols. While primary detection of wires is domain independent, it may and should be aided by higher level phases.

At the intermediate, syntactic level, we are already faced with a considerable degree of specialization, which makes it necessary to take into account the dual nature

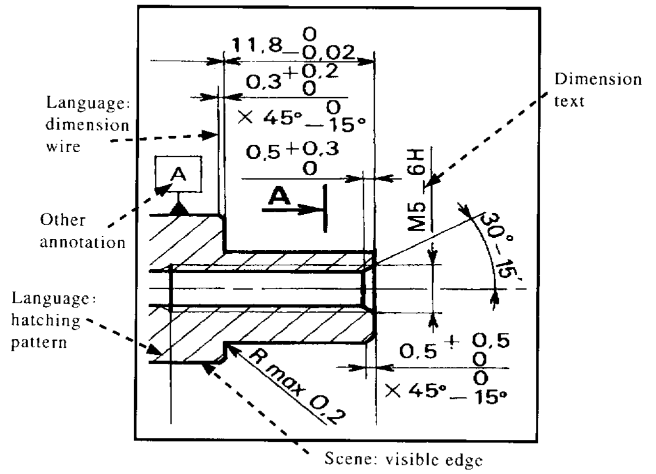


Figure 1 Illustration of dual nature of drawings

of drawings. *Figure 1* illustrates this point:

- Some of the wires belong to the image ('scene') part of the drawing: they correspond to a visible edge in the orthographic projection. At the same time, other wires are tokens of the 'language' part. For instance, some straight lines may belong to a crosshatching pattern, indicating a secion in matter, while other wires may be part of the dimensioning.
- Even the text layer cannot be treated uniformly: much of the text is associated with witness lines and tails of arrows, which together constitute dimension sets, while other text strings belong to different kinds of annotations, such as manufacturing instructions, or even text that is part of the product itself (like a control panel of a vehicle or any other sophisticated machine).

At the high, semantic level, the meaning of the aggregation of primitives is totally domain-specific and must be guided by *a priori* knowledge, not only about representation rules, but also about the technologies related to the represented objects, devices or setups.

Geometry and annotation layers and their separation

In order to extract useful, high-level information from a drawing, it ought to be viewed as a dual, two-faceted entity composed of several layers. Unlike most current commercial CAD-conversion systems, in which the separation into layers is done on the superficial basis of 'graphics' *versus* 'text', the semantic-conveying separation we propose here is based on the *function* of the entities in the drawing, rather than on their appearance, as follows:

- The *scene*, *image*, or *geometry* layer includes projections of the product and, possibly, cross-sections.
- The *language* or *annotation* layer contains dimensioning and functional symbols, such as the cross in a ball bearing centre, threadings, surface quality, welding, manufacturing instructions etc.

There is a difference between the customary *physical* text/graphics layer separation, in which small connected components are classified as text, while line art is classified as graphics, and our *logical* or *functional* layer separation. For instance, we advocate the classification of text components into one of the scene or one of the language layers according to its function. Alphanumeric characters and accompanying characteristic symbols, which are part of a dimension set or product description, are part of the language or annotation layer, while text which is part of the product itself (plaques, control panels etc.) is associated with the scene, image layer. Analogously, parallel slanted lines used to indicate crosshatching, for instance, belong to the annotation layer, as they represent a section in matter.

The spatial arrangement of the annotation text in dimension sets is itself a 'language within a language'. It denotes the type of dimension, its nominal value, and, optionally, the associated upper and lower tolerances.

Although the two layers are totally different in nature, they are very similar in their appearance: both contain mostly wires, and text may appear in both. This makes functional layer separation a difficult task that must rely on syntactic and semantic considerations. This task of layer separation is a key to the entire drawing understanding process: only a successful separation of the layers enables there to be proper interpretation of the relationships that stem from their combination. This, in turn, yields all the information needed to describe the product's model as it appears in the drawing.

CURRENT STATE OF THE ART: WHY ARE COMMERCIAL CAD CONVERSION SYSTEMS STUCK AT THE LEXICAL PHASE?

There are a number of technical document imaging systems on the market. In addition to a scanner and some preprocessing tools, these systems usually provide physical layer separation (into graphics and text, as noted above), vectorization, graphics primitive extraction and modules for limited character and symbol recognition. We do not elaborate on the variety of methods proposed for low-level feature extraction, such as vectorization, graphic primitive recognition etc. Several surveys on the state of the art in this area have been written, of available commercial systems[4], and of tools and methods proposed by various research groups[5]. Rather, we provide some pointers to the most usual methods, in order to assess what methods are currrently available.

- *Separating text from graphics* is usually based on the analysis of the connected components of the document image. Different techniques can be used to group these components into higher-level entities such as text strings, small isolated symbols or graphics parts[6]. Tools such as the Hough transform allow the identification of strings in any orientation[7]. One of the problems with text and symbols is that they may be connected to the graphics, so that the usual layer separation techniques fail. Some local techniques have been proposed to separate touching characters from lines[8,9]. As argued, the solution should not be limited to text-from-graphics layer separation; it must take into account the logical layering of geometry and annotation.
- *Vectorization* is the conversion of the graphics part of the bitmap image described in the raster file into vector representation of the lines constituting these graphics. Although a large number of methods have been proposed, the most usual approach in commercial systems is based on skeletonization and polygonal approximation of the skeleton (see Reference 10 as an example among many other references). Skeletons are known to be very noise-sensitive, but commercial systems usually 'fine tune' their skeletonization in various ways to get acceptable results. Many other vectorization methods have also been proposed based on detecting the lines by somehow following the two opposite contours of each line. Our own methods REDRAW[6] and OZZ[11,12] are variants of this principle. Avoiding skeletonization has the advantage of increased efficiency, since massive pixel operations needed for thinning are substituted for by sparse pixel recognition. Thinning is a well researched problem for which numerous methods and algorithms have been proposed; this also makes vectorization by thinning a good case in point to demonstrate a generic protocol for performance evaluation of a family of algorithms aimed at carrying out the same function or solving the same problem[13,14]. Admittedly, the number of factors to be considered in evaluating vectorization in the context of tasks that are more complex, such as engineering drawing understanding, is by far larger than that required by thinning. This renders the task of evaluating much more complicated. However, the basic idea of the performance evaluation protocol of a family of algorithms is still a viable starting point.
- *Graphics primitives* are detected by many systems; the most usual primitives in technical documents in addition to bars are broken lines[1], circles and circular arcs[7,12], arrowheads[12], and crosshatched areas[15].
- *Symbol and character recognition* is included as a module in some systems. The characters in strings can be recognized using one of the many available optical character recognition (OCR) techniques. Technical symbols are more difficult to recognize, but various specialized methods exist for this purpose[16,17].

In spite of the relative success at performing low-level operations, none of the existing commercial CAD conversion systems make any attempt at high-level analysis of the primitives they detect. Rather, these systems remain at the lexical level, while, as argued, real CAD conversion requires further analysis at the syntactic and then the semantic levels.

The major reason for this developmental 'freeze' is that, while theoretical foundations for at least part of the syntactic and semantic analysis were laid several years ago, applying them in working systems has been extremely slow. Our experience is that perhaps a major reason for this procrastination in applying higher analysis levels is the lack of market demand for such sophistication. This, in turn, stems from lack of knowledge on the part of the users about the potential enhancements and human labour savings that can be introduced for the systems they use by incorporating higher levels of drawing understanding. As the chief executive officer of one of the scanner companies put it when asked if he would be willing to invest in developing their product in this direction, 'our commitment is to our share holders', meaning that, as long as there was no pressure from their customers, they would not be willing to invest in such esoteric developments. It is clear, however, that once a product that successfully implements higher drawing understanding levels is introduced into the market, it will have an enormous competitive edge over all the existing products, and compel the manufacturers of conversion systems to enter this arena. The 'catch' here is that to develop these advanced capabilities requires large investments in detailed design, coding and implementation, which are normally beyond what researchers in academic institutes can afford.

We do not claim, of course, that it is possible to automate the conversion process in its entirety, as noise and ambiguities that exist in real, frequently faded or worn drawings, as well as ambiguities in annotation, will require some extent of human intervention and decision making. Following are some examples of possible noise-related problems that may cause the system to misinterpret the binary inputs at the various interpretation levels:

- *Lexical phase:*
  - Random salt-and-pepper noise may cause two parallel thin lines that are too close to each other to be interpreted as a thick line.
  - Arrowheads may go undetected (misdetection) while other arrowhead-like regions may be falsely recognized as arrowheads (false alarms).
  - Short, thick arcs may be recognized as bars.
  - Short bars (as in broken lines) may be classified as parts of text boxes.
  - Text touching the graphics may be considered as random noise and lost, impeding subsequent text recognition.
- *Syntactic phase:*
  - Dimension sets may be combined from the wrong components.
  - Annotation primitives can be classified as geometry and *vice versa.*
  - The recognized dimension text may not be in agreement with the actual measured dimension.
  - Because of missing or misdetected primitives at the lexical level, recognition of higher level entities can be erroneous or missed altogether.
- *Semantic phase:*
  - The geometry primitives do not form a topologically feasible 2D and/or 3D structure.
  - The dimensioning is either incomplete or redundant, violating the standard's proper dimensioning requirements[18].

In spite of these and a large number of other possible mishaps, some of which we cannot even predict until we actually get to the stage where they occur, we believe that the lion's share of the effort involved in conversion can be automated, leaving to humans only those points that an automated system is incapable of resolving. It is our estimate that, in drawings that are not too problematic, over 85% of the routine labour can be automated. A major problem which still remains is that the system, rather than asking for human decision making, sometimes makes its own erroneous decisions, which may go long undetected. Hence, a good system is not necessarily one that automates the drawing understanding as much as possible. Rather, it is one that optimizes the level of automation to minimize recognition errors at all levels.

3D RECONSTRUCTION

In this section we discuss the variety of transformations that engineering drawings may undergo and the differences among these representations. We then turn to a model of drawing understanding as performed by a trained human and propose a scheme for automating this process.

Engineering drawing transformations

A designer can design a model of a product or part of a product using advanced CAD software based on either constructive solid geometry (CSG) or a 3D boundary representation (3D B-rep). In both cases, a complete model requires the specification of tolerances, either explicitly or implicitly. The CSG tree can be converted into a 3D B-rep through a process of CSG interpretation. A 3D B-rep is transformed into a set of 2D B-reps by projecting it orthogonally from a number of viewpoints (front, top, side...). 'Fleshing out projections' is the inverse process, in which a set of orthogonal projections is used to reconstruct the 3D B-rep or a CSG assembly. Dimensioning and tolerancing of each projection in the CAD software according to an established standard (usually ISO or ANSI) add the appropriate dimension sets[18,19]. Adding the necessary manufacturing and logistic annotations completes the CAD drawing, which can then be converted into a paper drawing by plotting it. On the other hand, scanning a paper drawing converts it into a raster ('electronic') drawing. The raster version of the drawing may also be obtained directly by using software to rasterize the CAD drawing, with no noise being introduced by plotting and/or scanning.

CAD, paper and raster drawings are thus three versions of an engineering drawing model, which may also be classified as either a manufacturing drawing or an assembly drawing.

Human performance and 3D CAD conversion

A trained human expert reconstructs the 3D geometry of the object (part or assembly) by combining in his/her mind the views (orthographic projections) of the object depicted in the drawing. However, prior to this reconstruction, the human expert is able to extract a lot of information from a single view:

- Each view undergoes a mental 'layer separation', in which pure geometric entities (the projection of object contours) are discriminated from wires belonging to the language and annotation part (e.g. crosshatching, dimensioning).

- Annotation is further analysed; dimension sets are first aggregated from their primitive components: wires, arrowheads, and text strings containing numerical values and characteristic symbols. The resulting dimension sets are then associated with the corresponding object contours. The prudent drawing reader then sketches out a gross estimate of the match between the nominal value conveyed by the text and the actual dimension as seen in the drawing. Then, references to nomenclature (part names) are recognized as such, manufacturing instructions are read and understood etc.
- As drawings convey a lot of semantic information in a symbolic way, the engineer can also extract a lot of clues about the *functionalities* of the represented setups by looking at a single view. In our work on CELESSTIN, for instance, we reproduced in an expert system the 'reasoning threads' of an engineer analysing the drawing of a speed reducer or a similar device: he or she first follows axis lines to delineate the main components of the setup[20]. The engineer may recognize some components directly from their conventional representation, but he can also perform technical reasoning, such as disassembling or kinematics analysis of the whole setup[21,22].

As we propose to develop the automatic capability of understanding mechanical engineering drawings from its current low/intermediate degree to a degree that gets close to the capability of a trained human expert, we have tried to infer from the process carried out by such experts what stages an automated system should be capable of performing. The current state of the art for 3D reconstruction from several views does not take into account all the information which can be extracted from the automated analysis of each single view. The existing techniques for 3D reconstruction (also referred to as 'fleshing out projections'[23,24]) are limited to matching the *geometry layers* of the different views, i.e. to reconstructing a B-rep[25–28] or a CSG assembly[29,30] by matching the boundary projections or the projected faces. All of these methods assume that a clean, idealized set of projections is provided, uncluttered by annotation or any other 'irrelevant' graphic entities, and free of any noise or uncertainties about the exact dimensions of the lines in the drawing. This, of course, may be possible when the starting point is a set of views generated on a computer, available as formatted data structures, but it is an unrealistic assumption when the reconstruction has to be performed from views scanned from paper and processed by some vectorization package. This is so for the following reasons:

- As argued, the digitizing, vectorization and post-processing introduce errors and uncertainties into the drawing. It is therefore not possible any more to exactly combine segments or facets from a number of views, as their measured sizes will frequently not match to a predetermined accuracy.
- Annotation of various sizes interferes with the matching process; we must therefore be able to perform the layer separation naturally done by humans. However, if we are able to extract and recognize the annotation, especially the dimensioning, this weakness becomes a strength, as dimension analysis will allow us to correct the errors introduced by the document image processing.
- A view contains not only geometric entities which can be matched by 'fleshing out projections'. In addition to the previously mentioned dimensioning within the annotation layer, it also contains graphic entities interwoven within the geometry, representing symbolic information. This is the case for instance with broken (hidden) lines, dash–dotted (axis) lines, and crosshatching (crosssection) lines, which must be recognized as such. Not only are these symbols useless for direct 3D reconstruction, but their presence also introduces noise in the matching process, rendering the 3D reconstruction task yet harder. Once again, this weakness becomes a strength if we are able, for each view, to extract all the available information conveyed by this 'symbolic' part of the graphics, and make proper use of it. Thus, for example, the detected presence of an axis of symmetry is a valuable aid in determining whether we have indeed recovered the expected symmetrical primitives around it.

For all these reasons, 3D reconstruction using only some 'fleshing out projections' scheme may seem to be an unachievable goal. However, we maintain that the loss of precision and the numerous interferences in real-life drawings can be largely compensated for by taking into account all the information extracted by higher-level analysis of each view. Then, we can realize a human-like drawing understanding process in which the following set of operations is performed:

- *3D reconstruction:* Combine the real projection lines taken from the geometry layer into a 3D structure.
- *3D structure rectification:* Correct and rectify the resulting 3D structure by means of the dimensions, which have been extracted from the annotation layer.
- *Functionality assignment:* Add the functional information extracted from each view.

This scheme will be further developed in the sequel.

HOW CAN WE GO BEYOND THE CURRENT STATE OF THE ART?

During the last few years, our two research groups have conducted research in several directions in the engineering drawing understanding area, for various kinds of documents: city maps, electricity and phone wiring schemas, and mechanical engineering drawings. In order to do that, we investigated various vectorization techniques[6,11]. Our ultimate goal, however, has been the construction of systems that perform high-level interpretation of the document for the purpose of a suitable conversion into a description in terms of a CAD database and entities from its library. In this context, vectorization can be seen as a mere new encoding of the image signal, denser and more efficient than a raster image, but only a preliminary step in a much more elaborate understanding process.

Being researchers with a background in computer vision and pattern recognition, we naturally thought of structural interpretation techniques, such as graph matching. One idea, for example, was to chain vectors in order to build more complex structures. However, as argued, it rapidly appeared that technical drawings have a 2-fold nature: they are both an image, in the usual meaning of the projection of a 3D object on a plane, and a language, i.e. a way of communicating some knowledge through the use of specific symbols arranged spatially in a particular way.

Simple vector chaining has failed, because, for instance, when trying to connect vectors at junction points in order to extract longer lines, we must take into account the fact that some vectors belong to the orthographic projection of some 3D surface, whereas others belong to the language: they are either witness lines, which are a part of a dimension set, or crosshatching lines, which constitute the labelling of an area as a section in matter. The image and language use wires (bars and arcs) as identical basic primitives. In spite of their morphological identity, each wire carries exactly one of two entirely different meanings. Nevertheless, the two are interwoven within each other in complex patterns. This complexity is at the heart of the drawing understanding problem, and its solution calls for an original and flexible combination of approaches.

Our experiences: MDUS and CELESSTIN

The Machine Drawing Understanding System (MDUS), currently under development[12] (see *Figure 2*), is designed to automate the process of converting mechanical engineering drawings into an accepted standard, such as IGES (the Initial Graphic Exchange Specification), or DXF, for exchange of graphic information among CAD/CAM systems. MDUS comprises three phases that correspond to drawing understanding levels discussed above: early vision (the lexical phase), intermediate vision (the syntactic phase), and high-level vision (the semantic phase). The understanding phase of MDUS is preceded by preprocessing steps that include scanning and optional enhancement. The early vision stage of MDUS extracts the main primitives that construct most engineering drawings: bars (straight line segments), circular arcs, arrowheads, and text boxes: regions where text is expected to appear. Bars and arcs are generically termed wires. Our wire recognition algorithms detect wire endpoints, width, and, for arcs, also centre. A set of 'sparse-pixel' algorithms[12] are used to recognize bars, arcs and arrowheads, which are applied in this order as separate modules. Bars are recognized by the OZZ (Orthogonal Zig-Zag) algorithm, arcs are recognized by the perpendicular bisector tracing (PBT) algorithm, and arrowheads by the self-supervised arrowhead recognition (SAR) algorithm. The common, sparse-pixel trait of these algorithms is that they avoid massive pixel addressing and processing without compromsing the recognition quality.

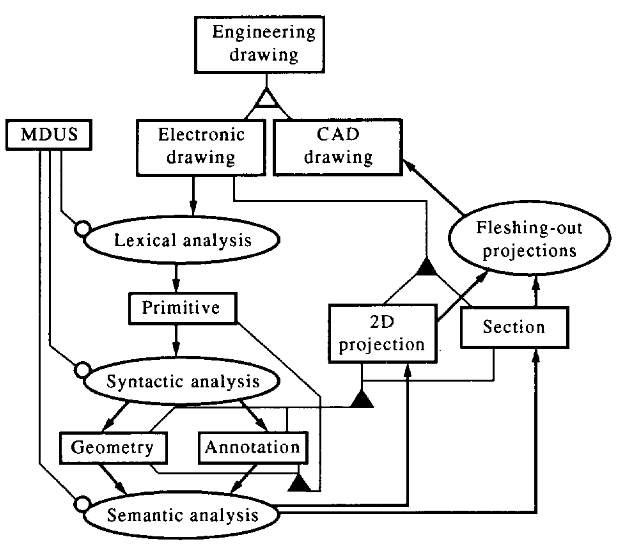


Figure 2 Overview of MDUS

Both of our groups have also been working extensively on the detection, recognition and analysis of dimensioning in engineering drawings. As noted, dimensioning is a language with a standardized syntax[31]; hence we can apply syntactic recognition methods to its analysis and recognition.

To gain insight into the syntactic process, we provide a most simplified description of the main syntactic operations in the process of dimension-set aggregation. Arrowheads and their attached tails are grouped into *leaders*. Leaders are matched both with each other to produce *leader pairs*, and with wires at which their arrowheads point to product the corresponding *reference*. Each reference can be either a *geometry site* or an optional *witness*, which guides to a *geometry site*. Leader pairs are also matched with their accompanying text boxes, which are located and extracted for further OCR. The aggregation of a leader pair, text box, 0 2 witnesses, and two geometry sites constitutes a dimension set. As noted, this is an extremely simplified description of the real process that should take place. Care needs to be taken of the dimension-set function (longitudinal, angular, radial etc.), symmetry (symmetric/asymmetric), arrowhead pointing direction (inward/outward), standard (ISO/ANSI) and more[31].

Specific tools are used to detect arrowheads and to segment the textual part as well as arrowheads[32]. The recognition and identification of the dimensioning is performed by using a 2D grammar; each dimension is then associated with the corresponding geometry site of the drawing, of which it becomes an attribute[33–35].

Figure 3 shows the result of the current performance of the lexical phase of MDUS on two engineering drawings. *Figure 3a* is the 300 dot/in^2 scan of the 'horseshoe' drawing. *Figure 3b* is the result of the primitive extraction, showing the bars, arcs and arrowheads detected in the drawing. All three arcs were detected, one in the 45° dimension set and two 180° concentric arcs in the geometry of the top view. To show that the arcs were detected, they are tinted. A fourth arc was detected as part of the digit 0 in the dimension text 160. Arrowheads are tinted inside a black frame. Note that, even though the original raster arrowheads are not quite identical, their parameters are detected and used for the representation in *Figure 3b*.

Figures 3c and *d* are another pair of the original 300 dot/in^2 raster of an 'oval hole' and its corresponding lexical phase output. Here, text boxes are also shown,

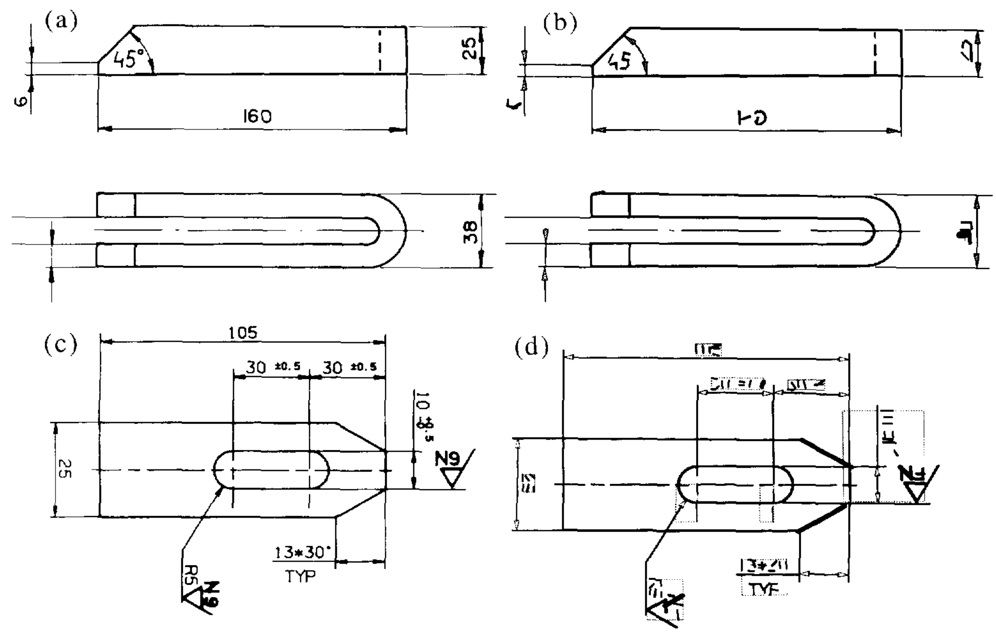


Figure 3 Results of dimensioning recognition

marked by dotted rectangles. Note that the rightmost detected box is larger than the real one, owing to the presence of extraneous short bars in the neighbourhood.

Overall, these results are encouraging and suggest that the subsequent syntactic and semantic phases have a satisfactory starting point. These phases, in turn, are expected to account for spurious or missing primitives found by higher-level considerations and cause the system to backtrack in search of misdetections and/or false alarms. Other teams[36–38] have also worked on dimensioning analysis and obtained similar results, confirming that this research direction is a promising one and should be further pursued. From 1989 to 1991 we also worked on CELESSTIN, an integrated, blackboard-based prototype system which converts drawings into a CAD description[39]. The first versions of this system were essentially based on structure and syntax to recognize entities such as shafts, screws, ball bearings or gears on a single view of a mechanical device. The system decomposes the vectorized document into a set of blocks having contextual attributes (hatching, threading etc.) and analyses these blocks by focusing on technical elements located along the axis lines. However, in the last version, CELESSTIN IV, we experimented with knowledge rules relative to the semantics: by focusing on a specific area of mechanical engineering, we were able to show that it is possible to analyse a single view of a drawing at the level of technological functionalities. We designed two 'experts', one focusing on *disassembling*, on the basis of the assumption that it *must* be possible to disassemble a mechanical setup, and the other dealing with the *kinematics* of the whole setup. The kinematics expert determines the functionalities of various entities from their behaviour when a rotation motion is applied around the identified axes in the drawing[22].

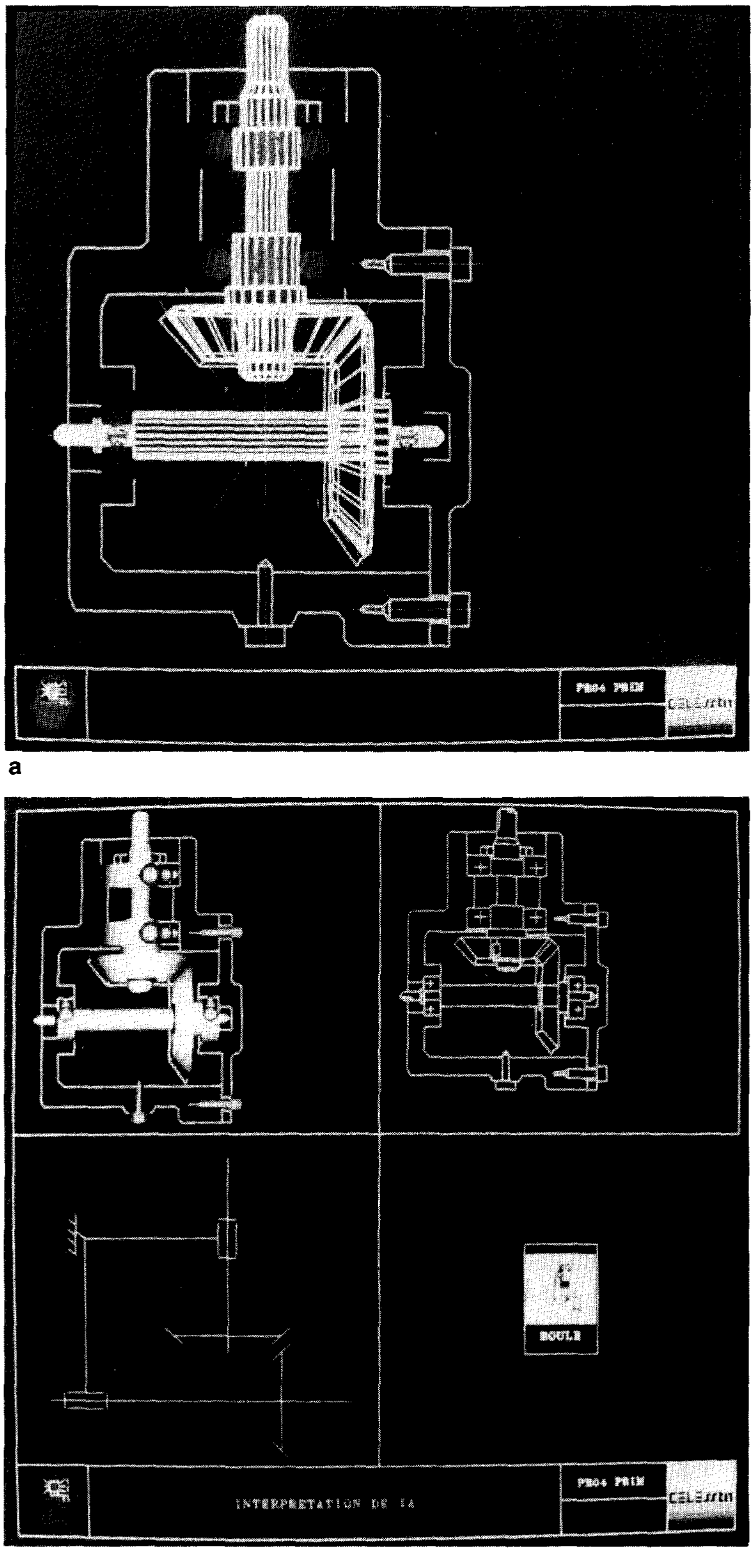
To illustrate this functional analysis, let us consider an example. *Figure 4* contains two screen dumps from CELESSTIN's functional analysis of a gearbox. *Figure 4a* displays the different functional entities the system has identified after disassembling and kinematics analysis, such as a gear, ball bearings, screws, and clips. *Figure 4b* provides another illustration of the same results. As these entities can be replaced by the corresponding entity in the CAD library, they can be displayed using the usual shading techniques (upper left corner); as the functionalities themselves have been identified, it is also possible to extract from the results of the interpretation the functional setup of the represented gearbox, as illustrated in the lower left corner.

Although we are aware of the fact that, even in the area of mechanical engineering, our prototype far from covers all possible functional interpretations, we believe that this work suggests a possible methodology for extracting functional information from technical drawings.

The experience we accumulated has convinced us that, for specific engineering domains, it is possible for automated document analysis systems to achieve the level of expertise of human engineers. More specifically, each single view of a drawing can be subject to lexical, syntactic and semantic analysis *before* it is combined with the other views for 3D reconstruction. This, in turn, opens the way to true semantic 3D modelling from scanned 2D views. In the rest of this section, we define more precisely our reasoning model and propose an agenda for achieving automated mechanical engineering drawing understanding.

'Hypothesize and test' in drawing understanding: a generic approach

Humans generate hypotheses based on the scene under examination and test them using evidence from any available source, be it from the same, a higher or a lower vision level. Hypotheses can thus be validated or negated 'on the fly' to produce a coherent, unambiguous geometrical and topological structure. This quest for evidence to help validate hypotheses generates *ad hoc* recognition loops as they are needed. As noted, regardless of the engineering domain and similarly to the compiler



a

b

Figure 4 Functional analysis by CELESSTIN

mode of operation, the understanding of the language part in engineering drawings consists of a lexical phase, a syntactic phase, and a semantic phase. Unlike compilers, though, in which the input is a 1D stream of unambiguous symbols, the input here is 2D rather than 1D. Effort must therefore first be directed towards recognition of these primitives. This process is evidently error-prone due to noise, overlap, and other difficulties in primitive recognition which present potential ambiguities. Thus, one cannot expect to obtain good results by just applying these three phases in series. Rather, a promising approach requires iterations from high level phases back to lower ones in order to test hypotheses that are aimed at satisfying sets of syntactic and/or semantic constraints.

In the domain of mechanical engineering drawings, dimensioning provides an exact specification of the geometry approximated by the contours of the orthographic projections. Dimensioning also specifies the tolerance around the nominal dimension for the part or assembly to be approved for further processing or marketing. Therefore, recognition of dimensions is a key component of MDUS. Syntax analysis is the basis for the 'parsing' of annotation in general and dimensioning in particular, but it is certainly not enough. This is so because at the lexical phase some primitives may either go undetected or be falsely identified. This prevents a complete association of each detected primitive with some higher-level entity, such as a dimension set, which, in turn, hampers sound semantic analysis.

To diminish the occurrence of such errors, we apply the 'hypothesize and test' paradigm, cycling around the various understanding levels. Compared with methodologies in general information systems analysis and design, ours is much like the 'fountain model'. This model calls for iterations from advanced stages in the development cycle of an information system, such as detailed design, back to earlier ones, such as analysis, in order to meet real-life constraints as they become apparent.

Applied to our situation, the fountain model stipulates that, rather than going linearly from lower to higher levels, we allow backtracking to check hypotheses generated in order to satisfy some constraints. This

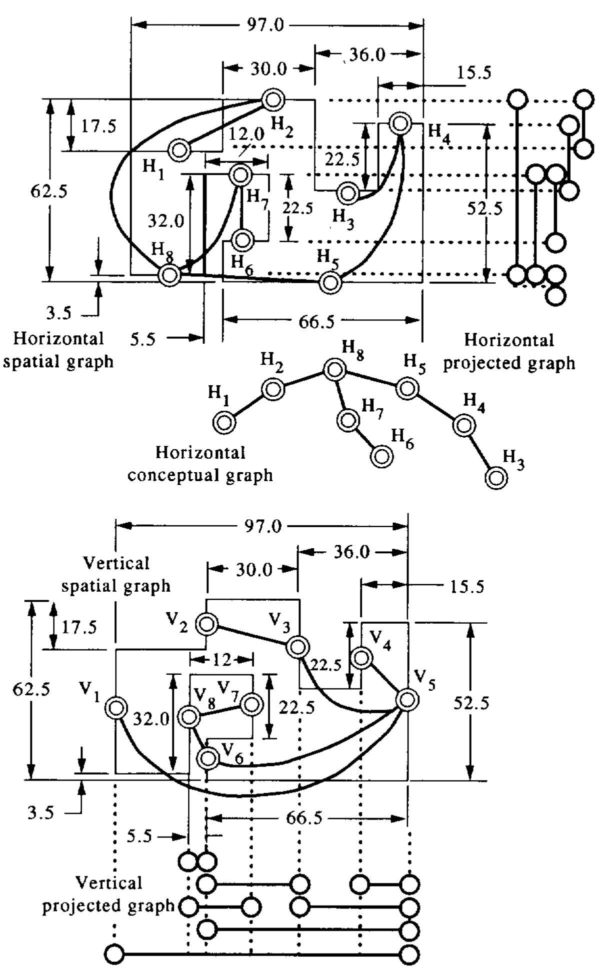


Figure 5 Horizontal (top) and vertical (bottom) dimensioning trees of same view

approach is opposed to the earlier waterfall model, which advocates linear, 1-way progress in the development process, and is analogous to applying lexical, syntactic and semantic analysis with no retreat option whatsoever. We hypothesize about and test for the existence of missing entities at all levels of abstraction, not just primitives. For example, during the semantic phase we take advantage of the *proper dimensioning theorem*[18]. It states the necessary and sufficient condition in order for a view to be properly dimensioned, i.e. have neither missing nor excessive dimension sets. The theorem uses a graph representation of the dimensioning of a view, as exemplified in *Figure 5*. According to the theorem, the dimensioning graph produced from representing each side in the view as a node in the graph and each dimension set between two sides as an edge connecting these nodes has to be a tree in order for the dimensioning to be proper.

Results from the semantic phase may not be consistent with the list of dimension sets produced by the syntactic phase, i.e. the dimensioning graph is, for example, disconnected (perhaps a forest). In this case, syntax-based search algorithms are called again to locate the missing dimension set where it is expected to be found, such that the corresponding graph cases to be disconnected.

What is on the agenda?

As we have shown, paradoxical as it may seem, the major obstacles in the drawing understanding problem, which is ultimately a 3D problem, are not in the 3D reconstruction. Rather, they have to do with proper 2D understanding of each projection separately. In this section we briefly summarize the major functions a drawing understanding system should be designed to carry out. These functions are listed in an increasing order of complexity.

- *Primitive recognition:* Enhance the existing state-of-the-art methodologies and develop additional tools for extracting primitives at the lexical stage. These primitives may be found directly on the bitmap image (arrowheads, small specific templates) or following the vectorization (crosshatching patterns, dotted, dot–dashed and broken lines etc.). Existing commercial systems, as well as algorithms developed by our groups such as OZZ[12], perform relatively efficiently and robustly vectorization and some basic template matching for extracting specific patterns. Circular arcs are also usually found by such systems, although errors still occur when discriminating between straight lines and circular arcs. This is a typical case where our proposed fountain model is useful. Consider an arc that is erroneously detected as a bar because of lack of sufficient curvature. When, at a later stage, the dimensioning analysis detects an angular dimension set and therefore expects a circular arc, the fountain model instructs the control algorithm to consult the raster source, this time with tighter parameters, to validate the hypothesis that the detected primitive is indeed an arc rather than a bar. More generally, the output of any vectorization scheme must not be considered as the 'plain truth' about the drawing. Rather, it should be taken as a first approximation of an exhaustive list of all the primitives with their correct attributes. This list is gradually improved and depleted as advanced recognition steps use its elements as building blocks for higher-level entities.

- *Syntax-based annotation analysis and layer separation:* Utilize the extracted primitives to obtain a coherent separation between geometry and annotation and aggregate dimension sets, including their textual content, i.e. the value of both the nominal dimension and the tolerance. Hand-printed character recognition can be verified by comparing the recognized numbers to the dimension values measured from the drawing itself, to alleviate basic mistakes in the character recognition process and to detect a mismatch between recognized text and measured dimensions. The expected output of this phase is a set of 2D orthographic projections along with the exact dimensions and tolerances. These will be the input for a 3D reconstruction scheme.
- *Functional analysis of each view:* Get as much semantic information as possible from analysis of the drawing based on the relevant application domain. This may include analysis of symmetries, kinematics of the setup or assembling schemes, or similar relevant features for the domain under consideration. Functional analysis pertains also to recognition of technologically meaningful entities from their symbolic representation, such as a ball bearing, represented by a cross, or a top-viewed threading, represented by two concentric circles. Other possible functional clues include the division of an assembly into groupings of parts which can be mounted or dismounted as a whole[40]. The recovery of this information for each view opens up new opportunities for faster and more reliable 3D reconstruction. For instance, if the system has recognized a ball bearing or a gear in one view, it can be instructed to look for it in another view, without the need for tedious, time-consuming and error-prone spatial 'blind search'. On the other hand, if the check of the other view reveals something completely different from what was expected, the fountain model of reasoning leads to yet another 'consulting session' with the functional analysis module.
- *3D reconstruction:* As dimensioning analysis provides the exact dimensions and tolerances associated with each view, it is possible to apply a 'fleshing out projections' process to the set of 2D orthographic projections and their associated dimensions to obtain a 3D boundary representation of the object described in the drawing. The edges of the object can be set to their true dimensions during this process. However, in order to prevent redundancies from one view to the other, each view is not completely dimensioned[18]; the correction of dimensions must therefore be done in parallel with the matching process, using constraint propagation techniques[41]. The matching of the geometric constraints must also take into account the language part. Therefore, the reconstruction operation has to integrate both state-of-the-art geometric matching techniques and the output from the knowledge-based interpretation of each single view at the semantic/functional level. *Figure 6* summarizes the approach we propose for 3D reconstruction. The drawing at the left hand side (which is incomplete and is meant only to make our point) is analysed in various ways, as explained in this paper: the raster scanned drawing is vectorized, text is extracted and recognized, the dimensioning is analysed, 'syntactic' features, such as crosshatching patterns, are identified and functional information such as the presence of a gear is extracted. The combination of dimensioning and the vectors can lead to the correction of the vectorization errors and hence to exactly dimensioned geometry, which can then be matched on the different views. This matching must be combined with the recognized functional and syntactic information to construct a true 3D CAD model. Throughout the whole process, the fountain model of reasoning is used and the different kinds of information are stored in a common database. In *Figure 6*, we suggest the use of a blackboard structure[42] as a means of implementing the fountain model, but other techniques are certainly also possible.

As argued, we do not expect to be able to carry out complete, generic 3D reconstruction for very complex

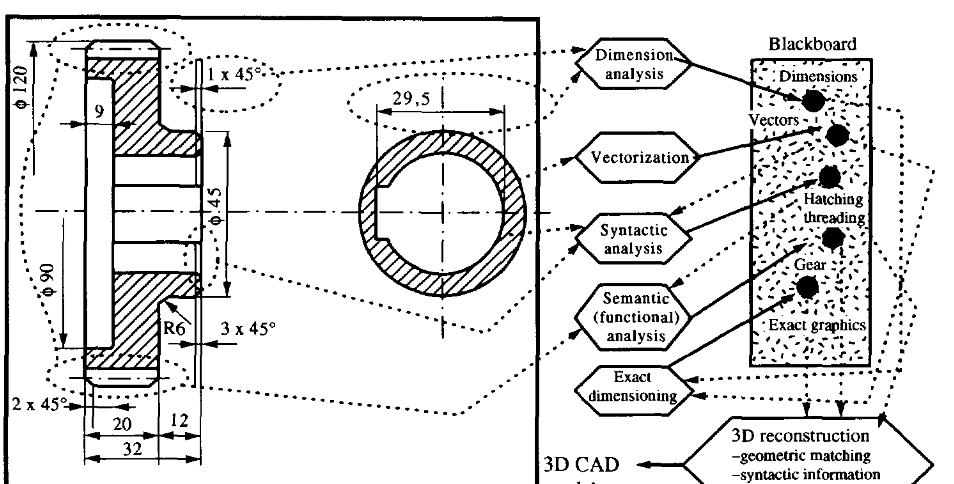


Figure 6 Principle of 3D reconstruction

drawings using these techniques alone. We do estimate, however, that it is possible to achieve this goal for drawings with low to medium complexity, within a well defined application domain, for which functionalities and semantics can be expressed using some kind of knowledge representation, and with a reasonable noise level. As more experience is gathered, we will be able to gradually tackle more complex drawings.

SUMMARY

We have described and analysed the need for high-level CAD conversion, the state-of-the-art in engineering drawing understanding, the challenge that this problem poses, and the difficulties that still lie ahead in the way of a comprehensive system for performing this task.

To advance in this direction, we propose a 'generate and test' constraint satisfaction strategy, which is based on the iterative nature of understanding complex scenes in general, and engineering drawings in particular, by expert humans. The approach is also inspired by the fountain model of systems analysis: rather than progressing in symbol interpretation linearly from the lexical through the syntactic to the semantic phase, we try to satisfy the sets of constraints imposed by the recognition, syntax and semantic analysis by cycles that are determined dynamically according to the scenario under consideration.

The significance of continued research in drawing understanding is 2-fold. In the specific domain of automating the understanding of mechanical engineering drawings, the expected prototype system will demonstrate the feasibility of closing the currently open loop between a raster drawing and its CAD drawing counterpart. This, in turn, will enable the development of a comprehensive system for converting paper and raster drawings into a CAD database, a feature which is in high demand from a large number of industrial organizations.

In the more general domain of visual knowledge representation and understanding within the fields of machine vision and artificial intelligence, the process of developing the mechanical engineering drawing understanding capability can serve as a case in point to formulate a generic theory of technical document understanding system construction. By comparing our work to works in other engineering domains, we expect to be able to characterize a generic system for technical document understanding and the process for developing such a system.

REFERENCES

1 Joseph, S H and Pridmore, T P 'Knowledge-directed interpretation of mechanical engineering drawings' *IEEE Trans. Pattern Anal. & Mach. Intell.* Vol 14 No 9 (1992) pp 928–940

2 Tombre, K 'Technical drawing recognition and understanding: from pixels to semantics' *Proc. IAPR Wkshp. Machine Vision Applications* Tokyo, Japan (Dec 1992) pp 393–402

3 Karima, M, Sadhal, K S and McNeil, T O 'From paper drawings to computer-aided design' *Comput. Graph. & Applic.* Vol 5 No 2 (1985) pp 27–39

4 Wolfe, L S and de Wyze, J 'An update on drawing conversion' *Comput. Aided Des. Report* Vol 8 No 6 (1988) pp 1–11

5 Kasturi, R, Raman, R, Chennubhotla, C and O'Gorman, L 'An overview of techniques for graphics recognition' *in* Baird, H S, Bunke, H and Yamamoto, K (Eds.) *Structured Document Image Analysis* Springer-Verlag, Germany (1992) pp 285–324

6 Antoine, D, Collin, S and Tombre, K 'Analysis of technical documents: the REDRAW system' *in* Baird, H S, Bunke, H and Yamamoto, K (Eds.) *Structured Document Image Analysis* Springer-Verlag, Germany (1992) pp 385–402

7 Kasturi, R, Bow, S T, El-Masri, W, Shah, J, Gattiker, J R and Mokate, U B 'A system for interpretation of line drawings' *IEEE Trans. Pattern Anal. & Mach. Intell.* Vol 12 No 10 (1990) pp 978–992

8 Joseph, S H 'On the extraction of text connected to linework in document images' *Proc. 1st Int. Conf. Document Analysis – Vol 2* Saint-Malo, France (1991) pp 993–999

9 Dori, D and Chai, I 'Extraction of text boxes from engineering drawings' *Proc. SPIE/IS&T Symp. Electronic Imaging Science & Technology, Conf. Character Recognition & Digitizer Technologies* San Jose, CA, USA (1992) pp 38–49 (SPIE Vol 1661)

10 Nagasamy, V and Langrana, N A 'Engineering drawing processing and vectorization system' *Comput. Vision, Graph. & Image Proc.* Vol 49 No 3 (1990) pp 379–397

11 Chai, I and Dori, D 'Orthogonal zig-zag: an efficient method for extracting lines from engineering drawings' *in* Arcelli, C, Cordella, L P and Sanniti di Baja, G (Eds.) *Visual Form* Plenum Press (1992) pp 127–136

12 Dori, D, Liang, Y, Dowell, J and Chai, I 'Sparse-pixel recognition of primitives in engineering drawings' *Mach. Vision & Applic.* Vol 6 (1993) pp 69–82

13 Jaisimha, M Y, Haralick, R M and Dori, D 'A methodology for the characterization of the performance of thinning algorithms' *Proc. 2nd Int. Conf. Document Analysis & Recognition* Tsukuba, Japan (1993) pp 282–286

14 Plamondon, R, Bourdeau, M, Chouinard, C and Suen, C Y 'Validation of preprocessing algorithms: a methodology and its application to the design of a thinning algorithm for handwitten characters' *Proc. 2nd Int. Conf. Document Analysis & Recognition* Tsukuba, Japan (1993) pp 262–269

15 Boatto, L, Consorti, V, Del Buono, M, Di Zenzo, S, Eramo, V, Esposito, A, Melcarne, F, Meucci, M, Morelli, A, Mosciatti, M, Scarci, S and Tucci, M 'An interpretation system for land register maps' *IEEE Computer* Vol 25 No 7 (1992) pp 25–33

16 Fahn, C S, Wang, J F and Lee, J Y 'A topology-based component extractor for understanding electronic circuit diagrams' *Comput. Vision, Graph. & Image Proc.* Vol 44 (1988) pp 119–138

17 Habacha, A H and Tombre, K 'Structural symbol recognition' *Proc. 7th Scandinavian Conf. Image Analysis – Vol 1* Aalborg, Denmark (Aug 1991) pp 486–493

18 Dori, D 'Dimensioning analysis: toward automatic understanding of engineering drawings' *Commun. ACM* Vol 35 No 10 (1992) pp 92–103

19 Dori, D 'Intelligent automatic dimensioning of CAD engineering machine drawings' *Int. J. Robot. & Automat.* Vol 5 No 3 (1990) pp 124–130

20 Vaxivière, P and Tombre, K 'Interpretation of mechanical engineering drawings for paper-CAD conversion' *Proc. IAPR Wkshp. Machine Vision Applications* Tokyo, Japan (1990) pp 203–206

21 Collin, S, Tombre, K and Vaxivière, P 'Don't tell Mom I'm doing document analysis; she believes I'm in the computer vision field' *Proc. 2nd Int. Conf. Document Analysis & Recognition* Tsukuba, Japan (Oct 1993) pp 619–622

22 Vaxivière, P and Tombre, K 'CELESSTIN IV: knowledge-based analysis of mechanical engineering drawings' *Proc. IEEE Int. Conf. Systems Engineering* Kobe, Japan (Sep 1992) pp 242–245

23 Preiss, K 'Constructing the solid representation from engineering projections' *Comput. & Graph.* Vol 8 No 4 (1984) pp 381–389

24 Wesley, M A and Markowsky, G 'Fleshing out projections' *IBM J. Res. & Develop.* Vol 25 No 6 (1981) pp 934–954

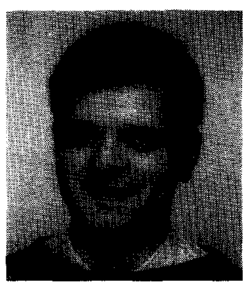
25 Lysak, D B and Kasturi, R 'Interpretation of engineering drawings of polyhedral and non-polyhedral objects' *Proc. 1st Int. Conf. Document Analysis – Vol 1* Saint-Malo, France (1991) pp 79–87

26 Martí, E, Regincós, J, Lopez-Krahe, J and Villanueva, J J 'A system for interpretation of hand line drawings as three-dimensional scene for CAD input' *Proc. 1st Int. Conf. Document Analysis – Vol 1* Saint-Malo, France (1991) pp 472–480

27 Kim, C, Inoue, M and Nishihara, S 'Understanding three-view drawings based on heuristics' *Proc. 11th Int. Conf. Pattern Recognition – Vol 1* Den Haag, Netherlands (1992) pp 514–517

28 Senda, T and Arimitsu, Y 'Automatic reconstruction of solid from a set of the orthographical three views' *Proc. IEEE Int. Conf. Systems Engineering* Kobe, Japan (Sep 1992) pp 229–233
29 Kitajima, K and Yoshida, M 'Reconstruction of CSG solid from a set of orthographic three views' *Proc. IEEE Int. Conf. Systems Engineering* Kobe, Japan (Sep 1992) pp 220–224
30 Tomiyama, K, Nakamua, T and Koezuka, T 'Auxiliary lines in three view drawings for 3D shape reconstruction using CSG method' *Proc. IEEE Int. Conf. Systems Engineering* Kobe, Japan (Sep 1992) pp 250–256
31 Dori, D 'A syntactic/geometric approach to recognition of dimensions in engineering drawings' *Comput. Vision, Graph. & Image Proc.* Vol 47 (1989) pp 271–291
32 Collin, S and Vaxivière, P 'Recognition and use of dimensioning in digitized industrial drawings' *Proc. 1st Int. Conf. Document Analysis – Vol 1* Saint-Malo, France (1991) pp 161–169
33 Dori, D 'Syntax enhanced parameter learning for recognition of dimensions in engineering machine drawings' *Int. J. Robot. & Automat.* Vol 5 No 2 (1990) pp 59–67
34 Dori, D 'Self structural syntax-directed pattern recognition of dimensioning components in engineering drawings' *in* Baird, H S, Bunke, H and Yamamoto, K (Eds.) *Structured Document Image Analysis* Springer-Verlag, Germany (1992) pp 359–384
35 Collin, S 'Syntactical analysis of technical drawing dimensions' *Advances in Structural and Syntactic Pattern Recognition: Proc. Int. Wkshp. Structural & Syntactic Pattern Recognition* World Scientific, Singapore (1992) pp 280–289 (Machine Perception and Artificial Intelligence Series Vol 5)
36 Lai, C P and Kasturi, R 'Detection of dimension sets in engineering drawings' *Proc. 2nd Int. Conf. Document Analysis & Recognition* Tsukuba, Japan (1993) pp 606–613
37 Lai, C P and Kasturi, R 'Knowledge-based understanding of engineering drawings' *Technical Report CSE-93-005* Dep. Computer Science & Engineering, Pennsylvania State University, USA (Sep 1993)
38 Min, W, Tang, Z and Tang, L 'Recognition of dimensions in engineering drawings based on arrowhead-match' *Proc. 2nd Int. Conf. Document Analysis & Recognition* Tsukuba, Japan (1993) pp 373–376
39 Vaxivière, P and Tombre, K 'Celesstin: CAD conversion of mechanical drawings' *IEEE Computer* Vol 25 No 7 (1992) pp 46–54
40 Lee, I, Dan, S, Kitahashi, T and Abe, N 'A study on a method of dividing machine-parts into functional groups for technical illustrations' *Proc. 2nd Int. Conf. Document Analysis & Recognition* Tsukuba, Japan (1993) pp 886–889
41 Tsujio, S, Ono, T and Lee, S S 'Computer-aided drawing check for CAD systems – a method for the checking of dimensions in multiview mechanical drawings' *Proc. IEEE Int. Conf. Systems Engineering* Kobe, Japan (Sep 1992) pp 234–237
42 Nii, H P 'Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures: Part 1' *AI Magazine* Vol 7 No 2 (1986) pp 38–53

Dov Dori is a senior lecturer in information systems engineering at the Faculty of Industrial Engineering and Management, Technion, Israel Institute of Technology. His research interests include machine vision, understanding engineering drawings, groupware, object–process analysis, and design and programming methodologies. Dr Dori received a BSc in industrial engineering and management from the Technion in 1975, an MSc in operations research from Tel Aviv University, Israel, in 1981, and a PhD in computer science from the Weizmann Institute of Science, Rehovot, Israel in 1987.



Karl Tombre is a research scientist at the Institut National de Recherche en Informatique et Automatique (INRIA), and is the coleader of a research group common to INRIA Lorraine, France, and to CRIN/CNRS, Nancy, France. He is working on computer vision, image analysis and document image interpretation. His main personal research activity is within the area of interpretation of engineering drawings. Dr Tombre holds a PhD in computer science from the Institut National Polytechnique de Lorraine, France. He has been employed by INRIA Lorraine since 1987. He is the chairman of IAPR Technical Committee 10 (Graphics Recognition).