Understanding Engineering Drawings: A Survey

Tapas Kanungo, Robert M. Haralick and Dov Dori

Intelligent System Laboratory
Department of Electrical Engineering, FT-10
University of Washington
Seattle, WA 98195

Abstract

Mechanical design and manufacturing information for 3-D solid objects has been effectively conveyed through a set of annotated orthographic projections and optional cross-sections. This forms the basis of engineering drawings, which solve the problem of unambiguously representing a 3-D object on a 2-D plane.

In this paper we address the inverse problem: given an engineering drawing of an object, construct the object's 3-D representation. To enable automatic recognition, the paper line drawings are initially scanned, and yield images which are inherently noisy. The 3-D objects themselves can have surfaces that are planar, spherical, or cylindrical. We examine the stages of drawing generation and formulate the drawing interpretation problem.

Most 3-D reconstruction algorithms have assumed that the vertex coordinates and line and arc endpoint coordinates are known accurately and without error, and that no annotation exists in the drawing. In practice, however, scanned drawings are noisy and contain annotation interwoven with the geometry lines. Current bottom-up rule-based systems do not utilize prior knowledge of the constraints imposed on 3-D object models, neither do they model the document degradation. Moreover, no performance evaluation of the systems for varying noise levels and object complexities has been carried out.

We compare the merits and drawbacks of the strategies employed in key works in the area of CAD model interpretation from engineering drawings, and propose research directions to enhance the practicality of paper engineering drawings-to-CAD conversion systems.

1 Introduction

Mechanical design and manufacturing information for 3-D solid objects has been effectively conveyed through sets of engineering drawings – annotated orthographic projections and optional cross-sections. The inverse problem involves the 3-D reconstruction of given line drawings of multiple views of an object. The line drawings are represented as scanned, binary images, and the objects themselves can have surfaces that are either planar or spherical or cylindrical.

Until the late 80's, this problem was largely considered as a geometry problem, in which line drawings were represented in terms of symbolic vertices and lines forming the orthographic projections of the object's wireframe. The exact 2-D projection coordinates of the vertices and line-endpoints were assumed to be known. It was further assumed that the input had no noise: no lines or vertices can be missing, and no extraneous vertices or lines can be present. Many algorithms were published [WM81, HQ82, Pre84, Ald83, SG83] to solve this problem for objects of varied complexity. Typically, these algorithms exploit geometrical constraints to prune the search space of possible 3-D objects.

More recently, this problem has begun to be addressed more realistically as a computer vision problem or, more specifically, as a document understanding problem, where the line drawings are represented as noisy, scanned images rather than symbolically. Here, the vertices and lines are not represented explicitly. In fact, some of them might be missing, while extraneous ones might be randomly present due to noise, which may have been caused by folding of the paper, stains, photocopier noise, etc. Typically, some initial image processing for noise cleaning and enhancement is typically performed, followed by primitive extraction. These primitives are then combined to constitute larger patterns which, in turn form the multiple views that are combined together to form a final interpretation of the 3-D object. The ultimate goal is to be able to represent the interpreted 3-D solid in a 3-D CAD format of a particular CAD system or a neutral CAD file format, such as IGES.

There are many reasons for the recent interest in this problem. Some of them were summarized in a recent paper [FF92] as follows.

- There are approximately 3.5 billion engineering drawings of various types in the United States and Canada, with about 26 million new ones added each year. The annual cost of filing, copying, accessing, and preparing these drawings for distribution exceeds one billion dollars.
- Once a drawing is in CAD format, all the advantages of database storage, retrieval and query become available.
- A major advantage of a CAD systems is that the time needed to modify a drawing is typically 13% to 33% of the time needed to accomplish the same revision using paper-and-pencil techniques.
- Only 13% (!) of the existing, active drawings are available in CAD form. In 1990, about 20% of the drawings were created in CAD form and about 25% were CAD revisions of older CAD drawings. The remaining 55% were done on the drawing board, using traditional paper-and-pencil drafting techniques.
- Cost-benefit analysis shows that if a drawing is expected to be modified several times, it is advantageous to convert it from hard copy to electronic format.

2 Engineering Drawing Generation Process and the Interpretation Problem Formulation

In this section we outline the process by which an engineering drawing of a 3-D object is created and the inverse problem of its interpretation. To design a good drawing understanding system, it is very helpful to have a model of the generation process of an engineering drawing. Following are the main features and stages of this process.

1. In many current CAD systems, an object is created and represented using a constructive solid geometry (CSG) tree. At this stage the information is symbolic – the object is represented in terms of union and subtraction of symbolically represented primitive 3-D shapes. An alternative CAD representation option for an object is by its boundaries (BREP). When an engineering drawing is prepared manually, the 3-D structure of the object is in the designer's mind only and has no explicit representation elsewhere.
2. Orthographic projections of the top, side, and front views are created. At this stage, the information is still symbolic – projected lines are represented by their end points, circles by their radii and center coordinates, etc. Along with the data representing the primitives (lines, arcs, circles, etc.) visibility and silhouette edges are also stored.
3. A subset of wire pairs (bars and/or arcs) is selected such that specifying their dimensions and tolerances completely specifies the 3-D object's geometry. To meet the standard's proper dimensioning requirements, this subet has to be minimal [Dor92]. The information at this stage is still symbolic.

4. The annotation (dimension sets and other manufacturing-related instructions) is placed on the drawing plane according to either ANSI or ISO standard. The information here is semi-symbolic: a mixture of vectors and text strings. For example, in ISO the dimensioning text is written along a line in the dimension-set [Dor88] called leader, hence it can be at any orientation. In the ANSI standard, the dimensioning text is always up-right, regardless of the orientation of the leader. This is the phase in which manually prepared drawings are completed, while the previous ones are in the designer's mind.

   To start an automated drawing understanding process, the drawing must be scanned to obtain its image. Scanning is a process that inherently introduces noise. Possible degradation introduced by the scanning and photocopying is due to blurring by the optical point-spread function, contamination by pixel-independent speckle noise, thresholding, random rotation, etc. These transformations can be modeled and simulated and a binary image is then created.

   At this stage, the representation has been rasterized and the information is no longer semi-symbolic, rather, it is a binary image. The vertex locations, line-type information and the higher level knowledge are not maintained in any explicit computer format. The only way it can be retrieved is by understanding (either by a human or by a machine) of the semantics conveyed by the drawing's image.

The engineering drawing interpretation problem can now be formulated as follows. Let I be a degraded image of a line-drawing representing a set of orthographic views of a 3-D object W, represented as a CSG tree with primitive shapes belonging to the set P. P is a set containing spheres, cylinders, and parallelepipeds. Furthermore, let W be such that the number of annotated projections provided in I are sufficient to completely specify the object. Given I, the task is to to determine W.

The rest of this paper reviews the literature on 3-D CAD interpretation from 2-D orthographic projections and compares current works that attempt at solving the problem. Although there is a large body of literature on related areas, such as interpretation of maps, circuit diagrams flow charts, etc. from scanned images, we will only mention them here. For further exploration, the reader is referred to a recent bibliography on document understanding by Kasturi and O'Gourman [KO92]. This bibliography also contains a section listing around thirty papers related to 3-D CAD interpretation that were published during the period 1986-1991. Nagendra and Gujar [NG88] reviewed papers published before 1988 on 3-D reconstruction from 2-D orthographic views. All the papers reviewed in that survey assume a set of noise-free orthographic projections as their input

3 Comparing Drawing Understanding Systems: Considerations and Criteria

In this section we discuss the considerations and criteria that are relevant in comparing the variety of existing drawing understanding systems.

3.1 Input

The nature of the input drawing and its content is a fundamental basis for our comparison. The following features are considered.

- Degree of polynomial surfaces representing the faces of the input objects. Drawings may represent convex polyhedral objects, non-convex polyhedral objects, non-planar faced objects in which faces are representable by second-order equations, etc., resulting in circular and elliptical arcs in the projection. Thus, the degree of the equation describing the most complex face in the object is a measure of the difficulty automated systems would have to face when required to recognize these surfaces. Most engineering drawings of medium complexity can be considered as containing only planar and cylindrical surfaces, which result bar and arcs in the projections. Most engineering drawings of medium complexity can be considered as containing only planar and cylindrical surfaces, which result in bars and arcs in the projections.

- Number of views needed/supplied. For some 3-D objects, only one or two views may be sufficient for a unique reconstruction. If only one view is given, a textual note must accompany denoting the width of the $2\frac{1}{2}$D object. This note, while easy to understand for humans, poses a severe difficulty to machine. For other objects, even three views many not be sufficient and there may be up to six views and any number of cross-sections to unambiguously define the object. The number of views and cross-sections is another feature of the drawing's complexity.

- Drawing standards. The drawing may conform to ISO, ANSI, or some other standard. It may also be a mixture of standards and contain "shortcuts" that can be easily understood by humans but challenges any "reasonable" automated system.

- Complexity of the annotation information. Annotation in an engineering drawing is divided into two major categories: dimensioning annotation and non-dimensioning annotation.

  Dimensioning comprises of a set of dimension-sets, each of which denotes the measure (length or angle) between two sites (geometry wires) in the object [Dor90]. Due to the proper dimensioning requirement, the number of dimension-sets is proportional to the complexity of the object as expressed by the number of its faces. The more complex the object, the more dimension-sets are needed to determine it. Since the white space on the drawing paper (and screen, for that matter) is limited, each additional dimension-set complicates the drawing in an increasing marginal fashion, since it interferes with the ones already drawn and cannot overlap or clutter any of the existing ones.

  Non-dimensioning annotation includes symbols for surface quality, welding, threading, bearing, textual manufacturing and finishing instructions, table of hole center coordinates for drilling, etc. A system that attempts to tackle real-life drawings should address these annotations and understand them at the highest level possible. This type of annotation is superposed alongside the dimensioning annotation and should not interfere with it.

- The uniformity of font and symbol. An original drawing which has been in use for some time may be marked-up by different engineers or draftsmen, each one with his/her own drafting habits and hand printing. This poses an additional degree of difficulty for a system that assumes a uniform style and scale. Moreover, it is frequently the case that changes in the dimension-set text are not accompanied by a corresponding correction of the actual drawn geometry. In this case, a system that is sophisticated enough to compare the recognized text with the measured value from the image, will find this discrepancy and prompt for awareness of this finding.

- The ratio of black-to-white pixels in the drawing. This ratio coarsely represents the density of the lines per unit area and possibly, the effect of noise. A ratio of over 0.05 is considered high.

- Adherence to standard. In engineering drawings the standard (ISO, ANSI, etc.) may not be strictly followed and shortcuts maybe present that pose difficulties to automated understanding.

- Symbolic noise-free versus scanned images. Symbolic noise-free inputs provide the system with the coordinates of the visible vertices and endpoints of the visible lines in each view. In the scanned images, the input is a binary/grayscale image, where no explicit information regarding vertex and line coordinates exists.

- Noise level. Scanned images are inherently noisy. Some low-level recognition algorithms are more robust than others with respect to pixel-noise. There is a variety of sources of noise, including the following.

- Degradation of the paper medium: originals and photocopies tend to fade over time. Their usage makes them stained, worn and torn and they may be glued by tape. The noise introduced may be in the form of straight lines (folding, tapes, etc.), arcs (coffee mug), and other irregular shapes (e.g., grease stains). The straight line noise requires a different treatment than the irregular noise. While the irregular noise can be treated by adaptive thresholding, lines are hard to remove automatically.
- Degradation due to photocopying and scanning. The optical process and the scanner process degrade the document further. For example, the optical process introduces blurring and speckle noise and the scanning process introduces quantization noise, skew, and random pixel noise. These processes have been modeled and simulated [Bai90, Bai93, KHP93, KHP94] and validated [KHB+94, KBH95].
- "Logical" noise. Lines or dimensions are missing or broken in the drawing. They can be easily restored mentally by humans, but the same operation is hard for machines.

- Input file format. The input file format may be a standard one (TIFF, GIF, Sun-raster) or pipelined directly from as scanner.
- Extent of human intervention. It is not expected that systems in the near future will be fully automatic. A certain degree of human involvement is necessary to resolve ambiguities and support the system's proper execution. The vast majority of the routine conversion work, though, is expected to be automated or else the system will not be cost-effective. The extent of the human intervention with respect to the level of drawing understanding is an important feature of CAD conversion systems.

3.2 Level of Drawing Understanding

The goal of a drawing understanding system is to convert an object represented on paper to CAD format. The output may be in IGES (neutral file) format or a format of a particular CAD system (Catia, Autocad, Medusa, etc.). The understanding of the system can be at different levels. Going from lower to higher levels, it has to perform bar recognition, but this is not sufficient. It should also recognize other primitives: arcs, arrowheads, textboxes. It may be able to recognize text, in which case it may be able to validate the recognition by an independent comparison to the values measured directly from the drawing. To do this, however, understanding at the syntactic level is required in order to extract dimension-sets.

Semantic understanding can be obtained at the 2-D level, for each projection separately, or at the 3-D level, where the interpreted 2-D views are combined to obtain the 3-D spatial description. These interpreted solids can be represented by CSG, BREP, or some other 3-D solid representation. Furthermore, if multiple interpretations exist, a line drawings, the system can produce all the interpretations or just the first one it finds.

Finally, 3-D kinematic understanding entails the capabilities of inferring the constrained motion or rotation of the mechanical set-up described in the original paper drawing.

4 The Phases of Drawing Understanding

Images of real engineering drawings are characterized by the following features: (i) They are in raster format resulting from scanning. Hence, information about the vertices, lines and faces is not explicit. (ii) They are noisy, so lines in the image may be broken and stray dots and lines might appear in the image randomly.

A complete drawing understanding process can be roughly broken up into three phases. Any drawing understanding system comprises of one or more of the following three phases.

4.1 Lexical Phase

The lexical phase starts with the noise reduction and is mainly the primitive recognition phase. Noisy images are restored and the basic constituents of engineering drawing are recognized. For references to some of the literature, please see the bibliography by Kasturi and O'Gourman [KO92].

4.2 Syntactic Phase

Here drafting rules and standards are embedded in a grammar and are used to check the correctness of the drawing. Correctness is checked with respect to syntax only, and not to feasibility of the drawn object. A typical problem may be that although the dimensioning follows the correct syntax, it may not make sense, but this will not be discovered until the semantic phase.

Dori [Dor92, Dor89] analyzed the contextual information provided by the dimensioning annotation in machine drawings. This context information can be utilized to resolve ambiguities. Extraction of dimensioning annotation from the drawing has been addressed by various researchers and is related to character and symbol recognition. Few other papers addressing the issue of understanding dimension annotation are by Dori *et al.* [DLDC93], Pao *et al.* [PLJ91], Fletcher and Kasturi [FK88], and Wahl, Wong and Casey [WWC82].

4.3 Semantic Phase

Here 2-D and 3-D understanding takes place. In the process it is checked whether the line-drawing represents a feasible object or not. Kinematic analysis may also be done at this stage.

The three phases need not be implemented in a feed-forward fashion. Feedback from other phases could serve for resolving ambiguities or for speeding up computation.

Early works have addressed the problem of 3-D reconstruction from 2-D projections, also know as the "fleshing out projections" problem. All these works completely bypass the first, lexical phase by assuming that perfect, "annotation-free" data of the 2-D projections (vertices, edges, faces, silhouettes, etc.) is provided.

Early work on interpretation of polyhedral objects from line drawings of one view of an object was done by Guzman [Guz68] and Waltz [Wal75]. The scenes could be viewed from arbitrary viewpoints, under orthographic or perspective projection and a relaxation labeling approach was used to solve the constraint satisfaction problem [HS79]. Sugihara [Sug86] gave the sufficient conditions under which a line drawing represents a feasible 3-D object. More recently Marill [Mar91], and Leclerc and Fishler [LF92] posed the problem of 3D interpretation from 2D line drawings as an optimization problem and gave results for noise-free line drawings. In the above papers the polyhedral scene was represented in terms of the vertices, lines, and faces, i.e., the information was provided in symbolic form as opposed to a binary, raster-scanned image.

Wesley and Markowski [WM80, WM81] gave a depth first search algorithm for interpretation of 3D polyhedral objects from 2D orthographic views. Haralick and Queeny [HQ82] posed the problem as a consistent labeling problem. A rule-based reconstruction algorithm for objects with curved (cylindrical, conical, toroidal and spherical) surfaces was proposed by Sakurai and Gossard [SG83]. Preiss [Pre84] used constraint propagation scheme to reconstruct 3D descriptions from 2D projections of objects with plane and cylindrical faces. Aldfeld [Ald83] proposed a pattern-matching scheme where 2D shapes are matched with hypothesized 3-D objects, until a consistent match is found. Most of the above papers have been discussed in greater detail in [NG88].

5 Toward Complete Systems: State of the Art

In this section we describe systems that are being developed. These systems start with noisy orthographic line drawings and produce 3-D solid descriptions. The first two systems (Langrana *et al.* and

Kasturi *et al.*) have high-level and the low-level modules working and are currently in the process of integration. Dori *et al.* have completed to low-level and part of high-level and are also in the process of integration. Joseph and Pridimore, and Vaxiviere and Tombre have systems that have all the stages working, but for a restricted set of objects. The system of He *et al.*, produces assembly plan from an assembly illustration.

Langrana *et al.* [NL91, CL92, NL90] use a volume-oriented approach to solve the reconstruction problem from noise-free projections. They assume that a complex object is composed of primitive objects belonging to a predefined class of 3-D objects and that these primitives can be recognized by making use of the knowledge of their typical 2-D projection patterns. Two examples of classes of objects handled by them are solids, obtained by a 3-D translation sweep operation and symmetric solids. Using these two classes, they can reconstruct plates, cylinders, parallelepipeds, wedges, spheres and cones. The system is similar to that of Aldfeld [Ald83]. It is implemented in Prolog and uses a heuristic search procedure to guide search. The system assumes that vertex and line information is noise-free, and does not handle multiple object assemblies. The system has the advantage of being interfaced to a real CAD system – MEDUSA. In [NL90, CL92] the authors give vectorization algorithms for real data, but have not yet combined their high-level and low-level systems.

Kasturi *et al.* [LK90, LK91a, LK91b, KBEM+90] are currently integrating two of their systems, one which handles the high-level interpretation, given lines and vertices of projections, and the other which finds the lines, vertices and annotations in scanned documents. The 3D interpretation system is similar to that of Sugihara [Sug86] but allows multiple views rather than only one view. They use a Dempster-Shafer formalism for the problem and their system can handle polyhedral, as well as curved objects.

Dori *et al.* [DLDC93] are currently developing a system called Machine Drawing Understanding System (MDUS), that takes as input a CAD drawing and converts it into an accepted standard for exchange of graphic information among CAD/CAM systems such as IGES (Initial Graphic Exchange Specification) [SW86]. It is assumed that the dimensioning follows the ANSI standards [Ame82]. Currently it can find bars, circular arcs, arrowheads, and textboxes and work is under way to construct the 3D description.

Joseph and Pridimore [JP92] use a top-down approach in their 3-D CAD interpretation system called Anon. Anon takes as input scanned 3-D CAD drawings with dimension annotations, and produces 2-D CAD descriptions. It does not handle multiple object assembly CAD drawings. It does not handle non-polyhedral objects either. The control structure is based in a psychophysical model proposed by Nieser.

Vaxiviere and Tombre [VT90] describe the system CELESTIN, developed in INRIA, France. It is a bottom-up rule-based system. The input to the system is a scanned 3-D CAD drawing without dimension annotations. The final interpretation is incorporated into the Catia CAD modeler, developed by IBM. Currently it cannot handle curved lines in the input CAD drawing, the input image has to be relatively noise-free, and the drawings have to conform to the French standards. Among the systems currently implemented, CELESTIN is one of the few systems that actually interfaces to a real CAD modeler ([NL91] is another). Furthermore, CELESTIN can handle multiple object assemblies in contrast to other systems which can handle only single object CAD drawings. The drawback of this system is that it is a flat rule-based system and the number of rules is large. This is due to the fact that they do not provide a top-level model. Rather, their model is implicitly hard-coded in the rules.

Automatic interpretation of assembly information was reported by He et al., [HAK92, HAK90]. Here the input to the system is a scanned image of an assembly manual page that pictorially describes how to assemble parts of a product. The task is to construct an assembly plan for automatically assembling the part. The main difference between assembly diagrams and manufacturing diagrams is that there there is hardly any dimensioning information in the assembly diagrams. The projections in the assembly drawings may be either orthographic or perspective.

6 Discussions

In this section we outline some limitations of the existing systems and the research approach. We then give some recommendations.

6.1 Limitations of current systems

Most systems fall into two very broad categories – the low level and the high level. In the low level category, there is a large body of literature on feature extraction and grouping of features into larger symbolic structures. But most of the current systems are stuck at this phase and have not been able to successfully reconstruct the 3-D solid from the extracted primitives. The reason for this state of affairs is that the current systems do not take into consideration the geometry of line-drawing formation process. That is, they do not have a model of the line-drawing image formation process. Thus, although there is a lot of information the current systems could have exploited by using a model-based approach, they resort to *ad hoc* techniques, where they have a flat rule-based reasoning system that tends to become very slow due to the large number of rules.

At the other extreme is the high level category with many theoretical and implementation papers describing 3-D reconstruction from noise-free data. Although these papers capture the geometry of orthographic projections, which the image processing-type systems have not, these papers do not at all consider missing and extraneous lines or loops. Thus these algorithms cannot handle real images the way they are.

Furthermore, there is a lack of performance measures and experimental protocols. That is, (i) the population of line drawings on which the algorithm 'works' is not well defined, (ii) the term 'error' is usually not defined, (ii) the number of images the reconstruction algorithm is not stated, etc.

6.2 Recommendations

First, there is a strong need for end-to-end systems. That is we need systems that take in degraded, scanned images and produce 3D shapes without human intervention. Furthermore, it is better to have a system that reconstructs simple 3D shapes from a large sample of images than a system that reconstructs one complicated shape from one image. Thus, more depth first (end-to-end) research is currently required as opposed to breadth first (complexity of objects, line drawings, etc) [DK93].

Second, criterion for evaluation and complete experimental protocols using the specified criterion should be used and reported [Har89, KJPH94, KJPH93]. This will enable the scientific community to replicate results reported in the literature. The performance evaluation should be based on a reasonably large set of simulated and real engineering drawings, which the system is supposed to process and understand. Many experimental systems work well on a small selected set of trial drawings but perform poorly over a large set. Different noise levels of various possible types should be tested to determine the robustness of the system and the noise level at which their performance is unacceptable. (See also section 3.)

We are in the process of creating a modest sized of line drawing images with ground truth for use by the line drawing community. At the DAS workshop in Germany this year there was interest for establishing protocols for evaluating performance of graphics recognition systems and as a result an international competition is being arranged at the conference to evaluate the performance of dashed-line recognition systems.

An approach to conducting controlled experiments is as follows:

1. Create 3D objects using a CAD modeler such as Autocad.

2. Generate the line-drawings of different views using the CAD modeler.

3. Generate the corresponding ground truth using the modeler (for example, Autocad provides the facility of producing the corresponding IGES files).

4. Create an ideal bitmap of the line-drawing.
5. Degrade the drawing either by using a document degradation model or by actually printing and scanning the document.
6. Run the reconstruction algorithm and compare the results with the ground truth. Note that if the 'real' degradation process is used in the previous step, the ground truth will have to be appropriately 'registered' before evaluating the reconstruction algorithm's output.

Furthermore, in order to evaluate the algorithm on manually drawn drawings, the same drawings can be created by a draftsperson using ink and paper, and then the reconstruction algorithm can be evaluated on these manually drawn line-drawings. In figure 1 we show an ideal line-drawings generated using Autocad and an artificially degraded version of the same drawing. The degradation was produced using our document degradation model [KHP94].

References

[Ald83] B. Aldefeld. On automatic recognition of 3d structures from 2d representations. *Computer Aided Design*, 15(2):59–64, 1983.

[Ame82] The American Society for Mechanical Engineers, New York. *ANSI Y14.5M, Dimensioning and Tolerancing*, 1982.

[Bai90] H. Baird. Document image defect models. In *Proc. of IAPR Workshop on Syntactic and Structural Pattern Recognition*, pages 38–46, Murray Hill, NJ, June 1990.

[Bai93] H. Baird. Calibration of document image defect models. In *Proc. of Second Annual Symposium on Document Analysis and Information Retrieval*, pages 1–16, Las Vegas, Nevada, April 1993.

[CL92] Y. Chen and N. A. Langrana. Restoration of cad database and geometric feature recognition. *Design Theory and Methodology*, DE-vol. 42:99–106, 1992.

[DK93] D. Dori and K.Tombre. Paper drawings to 3-d cad: A proposed agenda. In *Proc. of Int. Conf. on Document Analysis and Recognition*, Tsukuba, Japan, 1993.

[DLDC93] D. Dori, Y. Liang, J. Dowell, and I. Chai. Sparse-pixel recognition of primitives in engineering drawings. *Machine Vision and Applications*, 1993.

[Dor89] D. Dori. A syntactic/geometric approach to recognition of dimensions in engineering machine drawings. *Computer Vision, Graphics, and Image Processing*, 47(3):271–291, 1989.

[Dor92] D. Dori. Dimesioning analysis: Toward automatic understanding of engineering drawings. *Communications of the ACM*, 35(10):92–103, 1992.

[FF92] A. J. Filipski and R. Flandrena. Automated conversion of engineering drawings to CAD form. *Proceedings of the IEEE*, 80(7):1195–1209, 1992.

[FK88] L. A. Fletcher and R. Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:910–918, 1988.

[Guz68] A. Guzman. Decompostion of a visual scene into three dimensional bodies. In *Proc. of AFIPS Fall Joint Conf.*, volume 33, pages 291–304, San Francisco, 1968.

[HAK90] S. He, N. Abe, and T. Kitahshi. Undrstanding assembly illustrations in an assembly manual without any model of mechanical parts. In *Proc. of IEEE Int. Conf. on Computer Vision*, pages 573–576, Osaka, Japan, 1990.

[HAK92] S. He, N. Abe, and T. Kitahshi. Assembly planning based on assembly illustration understanding. In *Proc. of Int. Conf. on Pattern Recognition*, pages 61–64, Hague, Netherlands, 1992.

[Har89] R.M. Haralick. Performance assessment of near perfect machines. *Journal of machine vision and applications*, 2:1–16, 1989.

[HQ82] R. M. Haralick and D. Queeney. Understanding engineering drawings. *Computer Graphics and Image Processing*, 20:244–258, 1982.

[HS79] R. M. Haralick and L. G. Shapiro. The consistent labeling problem: part 1. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI–1(2):173–184, April 1979.

[JP92] S. H. Joseph and T. P. Pridmore. Knowledge-directed interpretation of mechanical engineering drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):928–940, 1992.

[KBEM+90] R. Kasturi, S. T. Bow, W. El-Masri, J. Shah, J. R. Gattiker, and U. B. Mokate. A system for interpretation of line drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):978–992, 1990.

[KBH95] T. Kanungo, H. S. Baird, and R. M. Haralick. Estimation and validation of document degradation models. In *Proc. of Fourth Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, Nevada, April 1995.

[KHB+94] T. Kanungo, R. M. Haralick, H. S. Baird, W. Stuetzle, and D. Madigan. Document degradation models: Parameter estimation and model validation. In *Proc. of Int. Workshop on Machine Vision Applications*, Kawasaki, Japan, December 1994.

[KHP93] T. Kanungo, R. M. Haralick, and I. Phillips. Global and local document degradation models. In *Proc. of Second International Conference on Document Analysis and Recognition*, pages 730–734, Tsukuba, Japan, October 1993.

[KHP94] T. Kanungo, R. M. Haralick, and I. Phillips. Nonlinear local and global document degradation models. *Int. Journal of Imaging Systems and Technology*, 5(4), 1994.

[KJPH93] T. Kanungo, M. Y. Jaisimha, J. Palmer, and R. M. Haralick. A quantitative methodology for analyzing the performance of detection algorithms. In *IEEE International Conference on Computer Vision*, Berlin, Germany, 1993.

[KJPH94] T. Kanungo, M. Y. Jaisimha, J. Palmer, and R. M. Haralick. A methodology for quantitative performance evaluation of detection algorithms. *IEEE Trans. on Image Processing (to appear)*, 1994.

[KO92] R. Kasturi and L. O'Gourman. Document image analysis: A bibliography. *Machine Vision and Applications*, 5:231–243, 1992.

[LF92] Y. G. Leclerc and M. A. Fishler. An optimization-based approach to the interpretation of single line drawings as 3d wire frames. *International Journal of Computer Vision*, 9(2):113–136, 1992.

[LK90] D. B. Lysak and R. Kasturi. Interpretation of line drawings with multiple views. In *Proc. of Int. Conf. on Pattern Recognition*, Atlantic City, 1990.

[LK91a] D. B. Lysak and R. Kasturi. Interpretation of engineering drawings of polyhedral and nonppolyhedral objects. In *Proc. of Int. Conf. on Document Analysis and Recognition*, pages 79–87, Saint-Malo, France, 1991.

[LK91b] D. B. Lysak and R. Kasturi. Interpretation of engineering drawings of polyhedral and nonppolyhedral objects from orthographic projections. Technical Report TR–91–092, Pennsylvania State University, 1991.

[Mar91] T. Marill. Emulating the human interpretation of line-drawings as three-dimensional objects. *International Journal of Computer Vision*, 6(2):147–161, 1991.

[NG88] I. V. Nagendra and U. G. Gujar. 3-d objects from 2-d orthographic views – a survey. *Computers and Graphics*, 12(1):111–114, 1988.

[NL90] V. Nagasamy and N. A. Langrana. Engineering drawing processing and vectorization system. *Computer Vision, Graphics, and Image Processing*, 49:379–397, 1990.

[NL91] V. Nagasamy and N. A. Langrana. Reconstruction of three-dimensional objects using a knowledge-based environment. *Enginering with Computers*, 7:23–35, 1991.

[PLJ91] D. Pao, F. Li, and R. Jayakumar. Graphic features extraction for automatic conversion of engineering drawings. In *Proc. of Int. Conf. on Document Analysis and Recognition*, pages 553–541, 1991.

[Pre84] K. Preiss. Constructing the solid representation from engineering projections. *Computers and Graphics*, 8(4):381–389, 1984.

[RV83] A. A. G. Requicha and H. B. Voelcker. Solid modeling: Current status and research directions. *IEEE Computer Graphics and Applications*, 2(2):9–24, 1983.

[SG83] H. Sakuri and D. D. Gossard. Solid model input through orthographic views. *Computers and Graphics*, 17(3):243–252, 1983.

[Sug86] K. Sugihara. *Machine Interpretation of Line Drawings.* MIT Press, 1986.

[SW86] B. Smith and J. Willington. *Initial Graphics Exchange Specification (IGES), Version 3.0.* U.S. Department of Commerce, National Institution of Standards, NBSIR 86-3359, 1986.

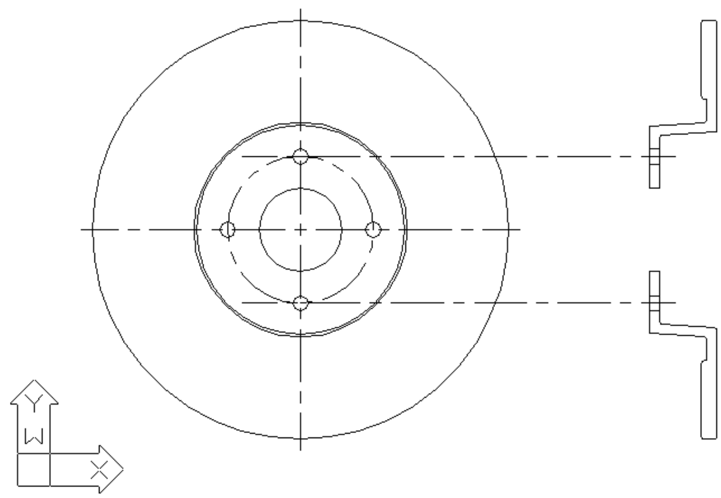
[VT90] P. Vaxiviere and K. Tombre. Cellestin: CAD conversion of mechanical drawings. *IEEE Computer Magazine*, 25(5):46–54, 1990.

[Wal75] D. A. Waltz. Understanding line drawings of scenes with shadows. In P. H. Winston, editor, *The Psychology of Computer Vision*, pages 19–91. McGraw-Hill, New York, 1975.

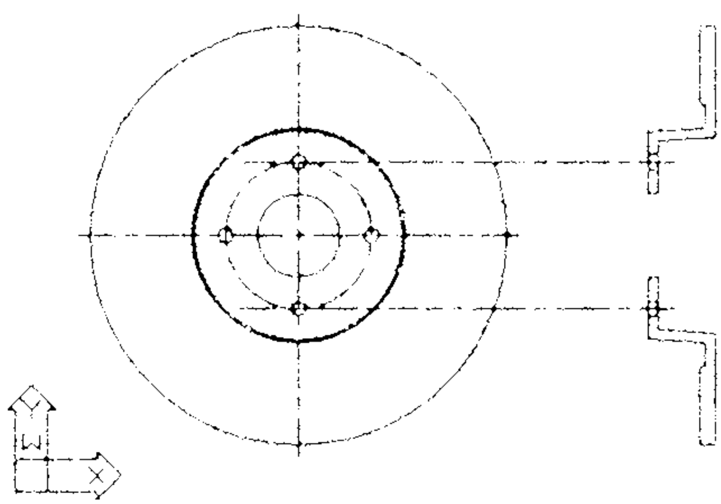
[WM80] M. A. Wesley and G. Markowsky. Fleshing out wireframes. *IBM Journal of Research and Developement*, 24(10):582–597, 1980.

[WM81] M. A. Wesley and G. Markowsky. Fleshing out projections. *IBM Journal of Research and Developement*, 25(6):934–954, 1981.

[WWC82] F. M. Wahl, M. K. Y. Wong, and R. G. Casey. Block segmentation and text extraction in mixed text/image documents. *Computer Vision, Graphics, and Image Processing*, 20:375–390, 1982.



(a)



(b)

Figure 1: An ideal line-drawing is created by Autocad is shown in (a). The corresponding IGES file can be created by executing one command in Autocad. The IGES file represents the ground truth for this line-drawing. (b) This binary image was created by artificially degrading the ideal line-drawing shown in (a). The degradation model used is described in Kanungo, Haralick and Phillips, 1994.