RECOGNITION OF ENGINEERING DRAWING ENTITIES:
REVIEW OF APPROACHES

S. Ablameyko*and S. Uchida**

*United Institute of Informatics Problems of the National Academy of Sciences,

Surganova 6, Minsk, 220012 Belarus. e-mail: abl@newman.bas-net.by

**Dept. of Intelligent Systems, Kyushu University,

Fukuoka-shi, 812-8581 Japan. e-mail: uchida@is.kyushu-u.ac.jp

Abstract

Recognition of engineering drawing entities is one of the most difficult stages in engineering drawing interpretation. There were made many attempts to recognize various types of ED entities. In this paper, we review algorithms for the recognition of ED entities, especially dimensions and crosshatching areas. For the recognition of dimensions, we analyze how dimension texts can be separated from graphics and how arrowheads of dimension lines are recognized. We also analyze the recent systems of ED interpretation. Finally, future tasks are discussed.

Key words: engineering drawing, image interpretation, vectorization, entity recognition.

1 Introduction

Automatic scanning and interpretation of engineering drawings (ED) started to be widely used to automatize their input into CAD systems. It allows one to transform drawings fast into digital raster form but usually requires a highly-developed software to convert scanned images into high-level models. The desirable output of this transformation for ED would be a 3D CAD model of an object drawn at ED projections.

Figure 1 shows a typical structure of ED interpretation systems. In this figure, the system is depicted as a diagram with three columns. At the left column, main stages are shown. Among those main stages, automatic stages are depicted as rectangles, and interactive stages are depicted

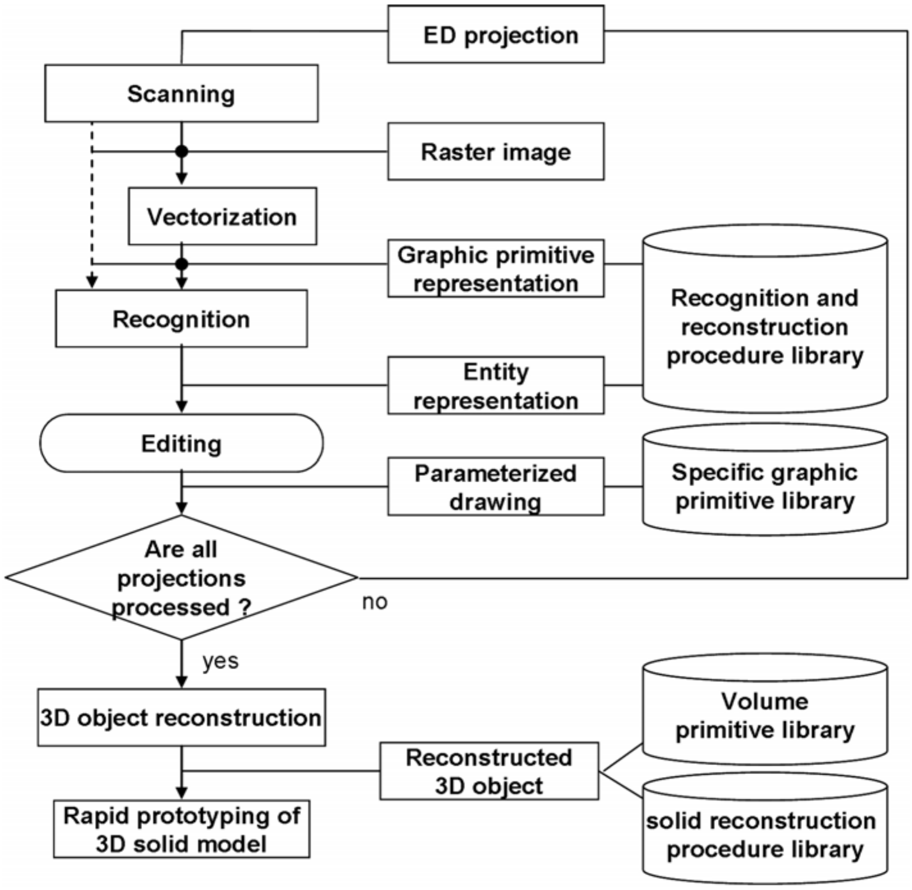


Figure 1: Diagram of ED interpretation system.

as circled quadrangles. The middle column shows image representations obtained after each stage. The right column shows procedures and libraries used at the stages. As shown in Figure 1, the ED interpretation system generally includes the following six stages:

1. scanning of the line-drawing to obtain a raster (binary or grey-scale) image,

2. vectorization of the raster image to obtain a vector image model in terms of simple graphic primitives (segments),

3. recognition of ED entities (such as closed contour lines, symmetry axes, hatched areas, dimensions, and texts) from the vectorized image,

4. interactive editing recognized drawing,
5. reconstruction of 3D engineering objects with all "semantic" attributes, and
6. rapid prototyping of 3D solid models.

In the rest of this paper, we firstly review algorithms for recognition of ED entities, which is one of the most difficult stages in ED interpretation. Among various ED entities, we focus our review especially on two difficult entities: dimensions and crosshatching areas. Furthermore, as important procedures in dimension recognition, we analyze how dimension texts can be separated from graphics and how arrowheads are recognized. Secondary, we review complete or near-complete ED interpretation systems. During the past 20 years, many ED interpretation systems have been developed for automatic or semi-automatic conversion of paper-based EDs to 3D CAD model. Some review material can be found in [70, 55, 66, 75].

We shall not touch in this paper a character (and other specific symbol) recognition task. There are many papers published for recognition of characters and symbols and we can refer only several review papers published recently [12, 20, 25, 49]. For editing and 3D reconstruction approaches, which are also omitted here, we can refer [9, 10, 13, 32].

2 ED entities

2.1 What are the ED entities?

Vectorized ED can be represented in two ways: by graphic primitives or by ED entities. The first representation is rather simple and not always useful. Straight lines, arcs, curves are the most commonly used graphic primitives. They can be grouped for forming more complex primitives, such as chained lines, text blocks, and arrows. Those complex primitives may be further combined into higher-level elements, that is, ED entities such as

- contour lines (often represented by solid lines),
- symmetry axes (dash-dotted lines),
- hidden contour lines (dashed lines),
- matter areas (represented by crosshatching),

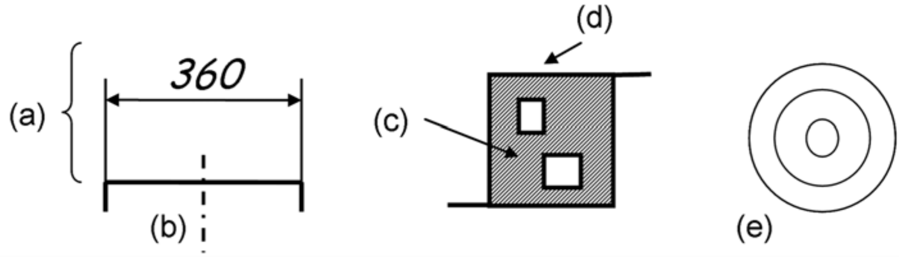


Figure 2: Examples of ED entities; (a) dimension, (b) symmetry axis, (c) crosshatching area, (d) border of crosshatching area, and (e) a set of concentric circles. Among them, (a), (c), and (e) are complex entities and (b) and (d) are simple entities.

- dimensions (thin lines with arrows, witness lines and so on),
- annotation text, and
- circles including concentric circles.

Examples of some ED entities are shown in Figure 2.

ED entities can be roughly separated into simple and complex ones. Simple ED entities are used for the description for simple ED structures, e.g., a symmetry axis, a crosshatching line, the border of matter area, etc., and can be represented as rather simple combinations of graphic primitives (or their pieces). In some cases, the simple ED entities can coincide with graphic primitives but in a general case they are more complex notions (for example, a symmetry axis can be represented by a poly-line consisting of a few dot-dashed straight lines and circular arcs, etc.).

The complex ED entities are used for the description of more complex ED structures, e.g., scenes, and can be presented as compound combinations of graphic primitives and a set of other simple and/or complex ED entities. Examples of complex entities are a crosshatching area (a set of hatching lines, bounded by one or more borders), a set of concentric circles, various dimensions, etc.

2.2 What is necessary to recognize ED entities

There are different methodologies of ED entity recognition. Some of them recognize ED entities directly from raster image representation [30, 40]. The others recognize ED entities from vectorized drawing [4, 26, 47]. In latter sections, we will be mainly concerned with the second group that is more commonly used in ED interpretation systems, while also showing algorithms from the first group.

To recognize ED entities, the following information must be available:

- parameters of entities to be recognized (thickness and length of segments, and of gaps for dashed lines, the slope angle for crosshatching lines, etc.);
- the structure of the entities, i.e., of what primitives the ED entity is composed;
- the spatial-logical relations between the primitives of one or different levels of representation and their meanings: joining or crossing under some angles; separation by a gap of some length; lie-in-neighborhood, and others. This knowledge can be effectively used for entity recognition.

Using the above information we can describe the entities either in a feature space (using only sets of parameters) or by means of a special formal language in a grammar form for the complex objects (scenes). That is usually done in most of ED entities recognition algorithms.

3 Recognition of dimensions

Dimensions are very important components of EDs; they provide high-level information that can directly influence our interpretation of EDs. Dimensions can appear in many forms. That is, the size, shape, orientation and style of dimensions can vary quite dramatically across a drawing. Most dimensions, however, generally follow a reasonably strict standard; they incorporate text and arrowheads. This means that the knowledge required to construct a dimension recognition system is in most cases quite easily available. A number of researchers have addressed this dimension recognition problem and most of conventional techniques dimension texts and arrowheads are firstly separated from graphics. Consider these tasks in more details.

3.1 Text separation from graphics in dimensions

The separation of text from graphics is a well-known and studied problem. Most of the solutions currently proposed, however, are intended for application to images of documents (pages of a journal, for example) in which text and graphics form distinct, independent, isolated zones. The situation is very different in ED images. Here, text and graphics are mixed and it is generally impossible to segment the image into the rectangular text-only and graphics-only regions assumed to exist in other areas of document image processing. It is noteworthy that a scheme for performance evaluation of graphics/text separation algorithms has been proposed in [72].

Text can be separated from dimension graphics in two ways: before vectorization, i.e. separation on raster image, or after vectorization, i.e. separation on vector image. Let us analyze published approaches starting from the first group.

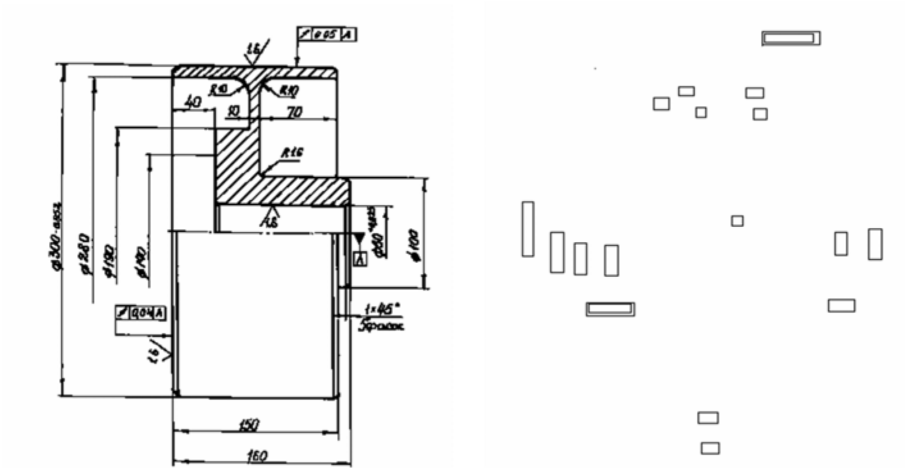
3.1.1 Text separation on raster image

A text separation algorithm based on connected area detection has been developed by Gao et al [39]. First, the algorithm finds all connected areas from drawing image. Then, a size criteria is used to find out character candidates from all connected areas and use a collinear criteria for grouping separate character candidates into text strings. Finally, text strings are analyzed according to text patterns summarized from dimension texts.

The basic principle of the algorithm presented by Lu [50] is to erase nontext regions from EDs, rather than extract text regions directly. To do it, analysis of connected components is performed and geometrical parameters of text characters are used. Nine parameters have to be set up that is not easy task. This algorithm can be used to extract both Chinese and Western characters and dimensions and has few limitations on the kind of EDs and noise level.

An approach for text/graphics separation using run-length-encoded image representation was proposed by Luo and Kasturi [52]. Directional edge images are all represented as directional run-length linked lists. As a result, both memory space and processing time are reduced significantly. The algorithm is independent of font style, font size, and language and map style. It performs well even in cases where lines and characters are connected.

A raster-based algorithm for text extraction in ED images was proposed by Ablameyko et al [4]. The contours of connected components are first extracted and some of their parameters (bounding



(a) (b)

Figure 3: (a) An image of a typical ED and (b) the text strings extracted from it.

rectangle, area and perimeter) are computed. On the basis of these values, potential text symbols are identified. Individual characters are then assembled to hypothesize text strings, depending upon the patterns of spacing between them. Figure 3 shows an ED and the text strings extracted from the ED by the algorithm.

A method presented by Dori and Wenyin [34] consists of three steps: growing individual character-box regions using a recursive merging scheme by stroke linking; merging the detected character-boxes into a textbox and determining its orientation; and re-segmenting the textbox back into the refined character-box that can be input to an OCR subsystem. The method handles both isolated and touching characters, aligned at any slant. The capability of segmenting characters that touch either themselves or graphics is obtained by focusing on intermediate vector information rather that on the raw pixel data.

In addition to those attempts, we can also find various approaches to separate text from graphics For example, morphology used in [51] is a promising approach.

The above raster-based algorithms can be roughly divided into two groups:

- algorithms based on finding character candidates by using information about their parameters like size for grouping separate character candidates into text strings;

- algorithms based on removal (e.g., erosion) of non-textual information that allow to leave only characters at the image.

3.1.2 Text separation on vectorized image

A vector-based method of segmentation and recognition of dimension text is presented by Dori and Velkovitch [33]. Its input is a set of vectors-wires (bars and arcs) resulting from the orthogonal zig-zag vectorization, arc segmentation, and arrowhead pair recognition. Initial textbox extraction is done by a region growing process, performed on text wire candidates. On the basis of textbox context (neighboring annotation wires), its drafting rule is detected. Raw textboxes are divided into logical textboxes, which are further decomposed into basic textboxes.

A related technique can be found in Yu et al. [77], where a vector-based algorithm is proposed to isolate symbols (e.g., symbols for electric devices) from lines in ED. The scanned image is thinned, vectorized, and then represented by a set of line segments that are input to the segmentation algorithm. Symbols and connection lines are separated using a few simple yet effective generic drawing properties, such as: (a) symbols contain mostly closed simple shapes, slant lines and open lines, and (b) connection lines contain mostly horizontal and vertical lines with each line end connected to a symbol. Connection lines are searched and detected first. Remaining entities are considered as symbols.

The above vector-based algorithms are mainly based on analysis of geometrical parameters of text characters and their components. Bounding boxes are often used that allow to extract and separate text from graphics. Several algorithms can process touched characters although the solution is not always robust.

3.2 Arrowhead extraction

The second step usually performed in dimension recognition is to locate and recognize arrowheads. Extracted arrowheads are often used as knowledge for starting dimension lines recognition [45, 1, 46].

Arrowheads may differ in shape, size, orientation and type. They are frequently obscured by image distortion, and/or noise. Interfering linework, i.e., occlusion by other lines also enhances the difficulty of arrowhead extraction [69].

The analysis performed by Dori et al. [30], however, suggests that the arrowheads are uniform in each mechanical ED, that is, they are usually all the same shape (triangular, rectangular or circular) and type (solid, hollow, stroke, wedge, half-filled or anchor) with in an ED. The self-supervised arrowhead recognition algorithm proposed in [30] takes advantage of this uniformity by breaking recognition into two phases: parameter learning and comprehensive search. The expected shape, type and size of arrowheads are estimated during parameter learning. The learned parameters are then used in the comprehensive search.

Dori [28] himself extended this approach to use a learning mechanism that deduces specific parameter values for a given ED and decreases the rate of false identifications. Syntactic considerations are employed to predict the spatial location and orientation of potential arrowheads that significantly decrease the search space. The parameter learning scheme may be applicable to a broader scope of tasks involved in intelligent recognition systems.

Priestnall et al. [58] have presented a method of recognizing arrowheads on EDs that takes advantage of a new data structure formed from the vector outlines extracted from scanned bi-level image. Opposite outline vectors are paired where possible, creating a structure of linked outlines representing linear features, and unpaired areas at junctions, corners and endpoints. This data structure is descriptive and preserves information, and allows rule-based feature extraction modules to be developed.

Min et al. [54] have proposed an arrowhead-matching algorithm for dimension recognition. Arrowheads are extracted by using a statistical method. In order to recognize contiguous dimensions, their algorithm forms an arrowhead chain, which is a sequence of arrowheads of the contiguous dimensions, and then matches two neighboring arrowheads on the chain. Finally, those paired arrowheads are used for dimension recognition. The paper have reported that algorithm can effectively recognize dimension subpatterns in Chinese EDs.

Recently, Wending and Tabbone [69] have proposed an arrowhead extraction technique robust to heavy occlusion. Their technique employs multiple criteria and combines them according to the Choquet integral concept which deals with a fuzzy evaluation.

An algorithm to extract arrowheads based only on their shape information was proposed by Ablameyko et al [1, 5]. A raster-scan approach is used in the approach. A line width histogram is computed and local maxima corresponding to thin and thick lines are identified. The line width is used as a parameter for pre-processing based on morphological erosion. A circular structuring

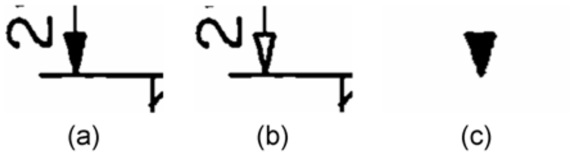


Figure 4: (a) An ED image, (b) extracted thin lines, and (c) an extracted arrowhead.

element is applied with radius equal to half the maximum thin line width plus one. Thin lines are deleted, leaving only arrowheads behind. These are then subtracted from the original image to produce a further image containing only thin lines (Figure 4 (a) and (b)). Arrowheads are then reconstructed, on the first image, forming isolated objects that are easily recognized by the distinctive shape of their contours (Figure 4(c)). If the image is expected to contain large black areas, further, similarly parameterized, erosion is used to delete thick lines. Dilation is then applied to reconstruct larger connected components.

Analysis of the above approaches for arrowhead extraction shows that most of them work with raster image representation. Algorithms take into account the specifics that arrowheads are usually much thicker than other ED entities. More simple approaches are based on mathematical morphology. For more difficult cases, learning phase and knowledge are introduced. Contour vectors can also be used to extract arrowheads.

3.3 Recognition of dimension structure

It was shown in many papers that dimensions can be described by a grammar, and a syntactical approach can be applied to recognize them. Dori and Pnueli [31] first described a dimension set as a conceptual web whose nodes are alternate components and associators. Dori has also extended this idea to other grammar-based approaches [27, 28, 29].

A plex-grammar formalism to define a dimension model was proposed by Collin et al [22, 23]. The dimension graphic primitives are extracted at its low-level stage. Assembly operators are then used to combine graphic primitives with one another. The dimension grammar is composed of a set of rules enabling the description of higher-level graphics. A set of different rewriting rules, called productions, gives the different possible definitions of the subshapes comprising the dimension to be recognized. Finally, all different dimension classes can be described: longitudinal dimensions, angular dimensions, or circular dimensions.

Collin and Colnet [24] have proposed a syntactic approach where a specific grammar is used to describe dimensions of EDs. Their grammar can be graphically designed by combining different graphic primitives. The algorithm used for analysis can start at different points of the grammar. The analysis proceeds bottom-up and top-down according to previously obtained results.

A system for detecting dimension sets in EDs was proposed by Kasturi's group [45, 15, 46]. A rule-based text/graphics separation algorithm based on analysis of connected components has been proposed. The procedure includes skeletonization, pruning, and arrowhead model generation and matching. After detecting arrowheads based on a model-based procedure, text blocks are extracted along with their corresponding leaders. Object lines are then separated from centerlines and hatching lines.

The approach proposed by Lin et al. [48] is organized in a three step manner; text/graphics separation, detection of arrowheads, and extraction of dimension lines. Text is separated from graphics in the thinned image based on the introduced index threshold. Then, feature points are identified and line tracking is performed. Arrowheads are extracted during recognition of dimension lines.

Su et al. [65] have proposed a dimension recognition method that starts by detecting potential dimension frames, each comprising only the line and text components of a dimension, and then verifies them by detecting the dimension symbols. The method is capable of handling low quality EDs.

A knowledge-based approach and multi-level scheme for dimension structure recognition was proposed by Ablameyko et al [34, 47]. The upper level is the dimension itself; the next levels are text, shape line, and extension line. In turn, these are subdivided as follows: text— into individual characters defining in some cases the type of the dimension, the magnitude of the dimension, and the tolerances; shape line—into a dimension line, arrowheads, and tails (prolongations of a dimension line), etc. The elements of each level are assigned descriptors, that is, the descriptions of the elements in the proposed language are represented, in the simplest cases, in terms of metric and topological features or, in the more complicated cases, in terms of their relations with other elements at the lower level of dimension representation. In addition, every dimension component is accompanied by a set of possible relations between a given component both with other dimension components at the same or higher level and with other elements of the ED in general. This

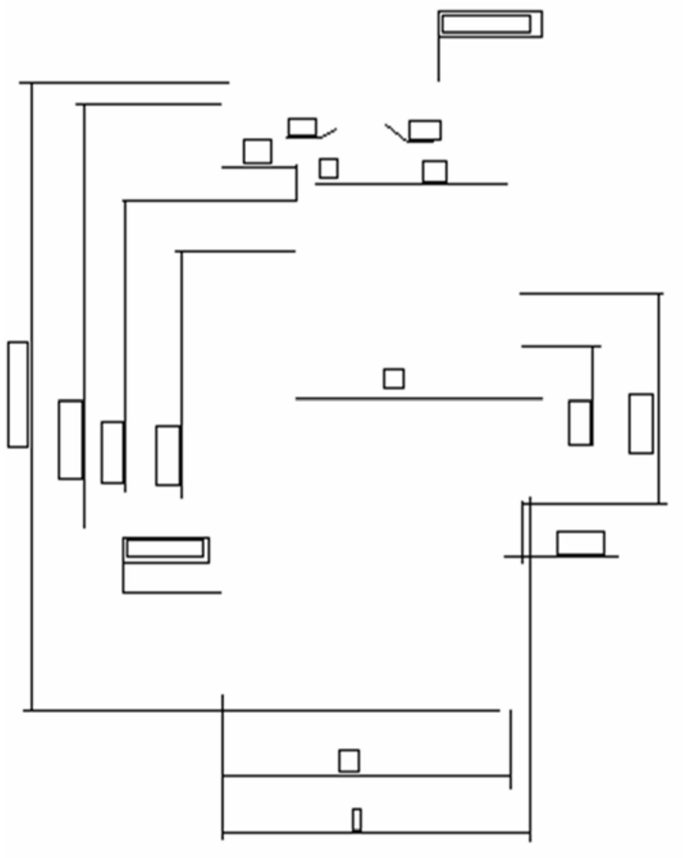


Figure 5: Dimensions extracted from the ED shown in Figure 3.

knowledge permits the use of all information about the drawing recognized so far, thus speeding up the interpretation and, in some cases, making it more reliable.

Figure 5 shows the dimensions extracted by [34, 47] from the ED of Figure 3. Experimental results in [34, 47] have suggested that the algorithm is capable of extracting dimensions from such images with a high level of reliability. This is because it exploits both a priori knowledge and the results of other entity recognition subsystems.

Analysis of approaches for dimension recognition shows that practically all of them use syntactical approach and formal grammars. The dimension graphic primitives are extracted at the low-level stage and their assembly is then used to form higher primitives. The dimension grammar

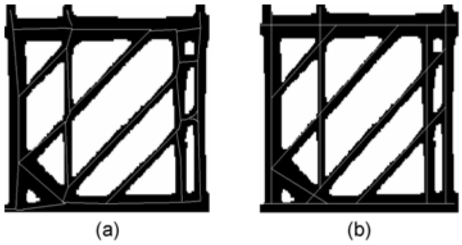


Figure 6: A crosshatched area and its vectorization; (a) before and (b) after defect reduction.

is composed of a set of rules enabling its description. Another peculiarity of dimension recognition algorithms is maximal use of dimension specifications stored in a knowledge-base. Many algorithms use arrowheads and dimension text that are extracted at the beginning stages and used further as starting points in dimension recognition.

4 Recognition of crosshatched areas

Crosshatched areas are common line drawing components representing sections on EDs. (Buildings on maps are also represented by crosshatched areas.) Although the styles of crosshatched areas obviously vary in their details, they are always a variant of parallel lines somehow gathered together to form closed regions.

Recognition of crosshatched areas is usually performed at vector representation obtained by thinning and vectorization. The thinning operation in such image types inevitably introduces defects, which badly affect the recognition process. Thus, analysis and improvement of thinning result should be done. It is usually done by operation that is called "pruning." There are several types of possible defects and approaches to reduce them [1, 11]. In the case of crosshatched areas, such important and difficult operations like line aligning, changing of positions and number of nodes should be made. Figure 6 shows a section of vectorized ED before and after pruning by the algorithm developed by Ablameyko et al [5, 11].

One of the first approaches to analyze objects containing regular set of lines (texture lines) was proposed by Kasvand [43]. He have studied a general case where no a priori information is available on the texture elements or on the shapes and number of lines. This generalized approach was computation intensive, but nearly all of the computations are of the form for which pipelined, flow, and parallel processors can be designed. Prior knowledge of the shapes of the lines or the nature of the texture elements can be used to bypass many of the computational steps, or the computations may be terminated earlier when sufficient information has become available for solving practical problems.

An interesting method was developed for the French systems CIPLAN, REDRAW, and CELESTIN [14, 68]. Elongated connected components with common width and orientation are first extracted from the raster image. These provide a low-level image model. The image is then vectorized and crosshatched areas are formed using information from both the vector database and the image model. The method relies heavily on a priori knowledge. All three systems adopt the same basic approach, but differ in their high-level processing.

In a system developed by Boatto et al. [17], crosshatched areas are recovered from raster images of land register maps. An image graph describing connected components and their interrelations is formed and used to guide the search of a closed polygonal area of the image. Any nodes and edges considered to form part of crosshatching lines are removed from the image graph. If the search for hatching fails, a human operator is asked to provide help.

A different-style algorithm proposed in [61] finds perceptually salient and compact closed regions in line drawings, such as EDs. This purpose matches with the task of detecting crosshatched areas. The algorithm tackles with the constrained optimal path finding problem on a graph constructed from an input line drawing. The Gestalt laws of perception underlies the formulation of the problem.

A vector-based algorithm to recognize crosshatched areas in EDs has been proposed by Ablameyko et al [7, 3]. The area is assembled on a vector image where all lines and area boundaries are detected concurrently. Thus the dependence on line width is reduced to a minimum. In addition, a-priori knowledge about the type of document and the rules for the construction of hatched areas in a drawing are used. Because the algorithm is a part of a system, it utilizes the information obtained at the previous stages of image processing. The information used is concerned with the type of assembled lines in terms of graphic primitives— straight lines, arcs, dashed lines, and line segments of the S-form.

The result of applying the processes described above to the image shown in Figure 3 is presented in Figure 7. The algorithm has been tuned to EDs and consequently produces good results on this

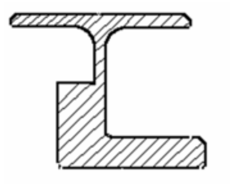


Figure 7: Crosshatched area extracted from Figure 3.

type of image.

Analyzing this task, one can say that there are less papers and attempts to recognize crosshatching area then for example to recognize dimensions. It can be explained that crosshatching areas are less spread then dimensions. From other side, this task is more complicated as we think. Most of the approaches are based on search of collinear lines and on extraction of border surrounding these lines. The proposed approaches are not robust in general and strongly depend on image quality.

Recognition algorithms must successfully cope with many of the defects and have a sufficiently high degree of noise immunity. Moreover, using the information obtained at every recognition stage makes it is necessary to keep the percentage of erroneous decisions to a minimum. Alignment and joining will level off many defects and the image takes on an appearance similar, in content, to the original.

5 Recognition of other ED entities

5.1 Recognition of dashed lines

The first step in entity recognition can be a detection of dashed and dot-dashed lines. In so doing, both the shape (as in the case of straight lines) and the structure (as in the case of arbitrary broken lines) of the lines in question are used. There are quite many algorithms for dashed line detection and many systems use those algorithms. In fact, a contest for dashed line detection was organized [36, 44, 71]. Performance evaluation protocols of the contest are described precisely in [73, 56, 57, 21]. The algorithm by Ablameyko et al.[4] for recognition of dashed and dot-dashed lines consists of the following stages: a) sequences of segments having local properties inherent in

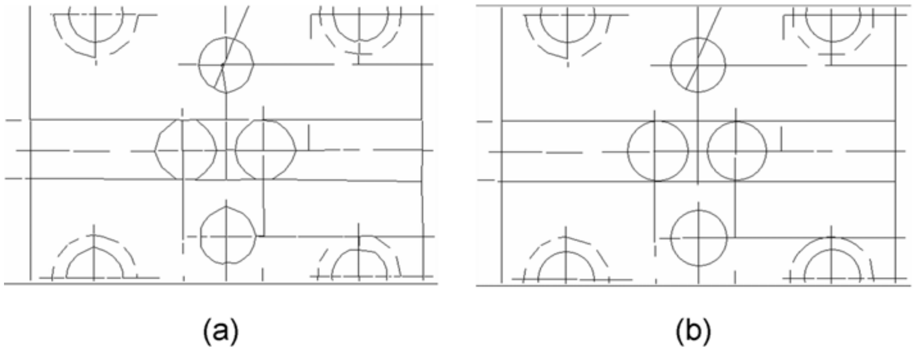


Figure 8: (a) A vectorized ED. (b) Arcs and circles are extracted and recognized.

a dashed or a dot-dashed line are detected; b) the extension of a line is sought in ambiguous cases (processing of gaps, nodes, etc.). Proceeding from the dot-dashed lines thus detected, the potential centers of circles are detected and, on their basis, arcs (both continuous and discontinuous) with labeled centers are identified.

5.2 Recognition of arcs

Arc extraction is also quite established and investigated topic and there are quite many published papers [60, 74]. For example, the approach in [38] is based on vectorizing a binary image, smoothing the vectors to a sequence of small straight lines, and then attempting to fit arcs.

However, if we need to extract circles this is a little more difficult problem. Circles may touch one another line and may pass through any given vector. In [4], Ablameyko et al. have developed an algorithm to extract arcs and circles from vectorized drawing. It is based on a modified least squares method and consists of two main stages:

- identification of graphic primitives, without an analysis of relations between them;
- joining and alignment of graphic primitives, subject to the relations between them.

The latter stage is very important for ED interpretation because most of lines to be extracted should be horizontal or vertical, and arcs should be as long as possible. Figure 8 shows the arcs and circles extracted from a vector representation of a section of ED.

5.3 Recognition of contour lines

In [8], an attempt is made to combine thick lines, which are not the boundaries of hatched areas, into the contours of closed areas. This operation produces contour lines, from which the filled areas represented by contour boundaries, if there are any, are assembled. The segments that make up a closed contour (block) are assembled in the following basic stages: a) the initial segment is chosen; b) the direction for tracing a closed polygon is determined; c) the chain of segments is traced in the direction thus chosen, and the path is analyzed.

The method proposed in [19] includes the following steps: 1) construct networks of single closed regions (SCRs) of black pixels with all the information about both segments and their linking points; 2) classify all the digital contours represented by SCRs into three types: straight –line segments, circular arcs, and combined lines; 3) decompose the combined lines into least basic sub-lines or segments with least fitting errors; 4) determine their relationships.

6 Post-processing after ED entity recognition

Once ED entity recognition is complete, geometric correction is usually performed. Near-vertical or near-horizontal lines become strictly vertical or horizontal and near-parallel lines become parallel etc. At the same time, points of contact between different entities are revised.

It should, however, be noted that the quality of the interpretation naturally depends upon the quality of an original drawing. If the input drawing suffers defects beyond those implicitly expected by the system, interactive post-processing or other operator support will be required [53]. As the current system produces standard CAD format files, interactive editing could be done within a CAD environment.

7 ED Interpretation Systems

7.1 Two approach of ED interpretation – Top-down and bottom-up approach

Current approaches to line drawing interpretation can generally be classified as either bottom-up or top-down [11]. The bottom-up approach is characterized by an emphasis on the analysis of small groups of connected or otherwise physically closely related pixels and relies upon data-driven, local processing. Bottom-up interpretation systems tend to start with the image and move towards abstract, entity-level descriptions. In contrast, top-down approaches concentrate on the relationships among graphical primitives, objects, and scenes. Systems built around this type of architecture typically begin with some description of the entities they expect to find and precede by seeking evidence for the presence of those entities in the input drawing. Such systems therefore rely more on model-driven, global processing. At present time, bottom-up approaches remain the most commonly used, particularly for lower level interpretation tasks such as map vectorization. Top-down approaches are usually referred to as knowledge-based because they use a priori knowledge to guide object recognition.

The simplest, and most commonly used, bottom-up line drawing interpretation system architecture is classic sequential structure. Figure 1 shows a diagram of the sequential structure. Systems based upon this model comprise a linearly ordered set of independent processes: each receives the output of its predecessor and passes its own output on to the next process in the chain. The drawing image is input to the first and the interpretation emerges from the final process. We should note that this bottom-up movement is not straight away. In some cases, several passes through the technology should be done [11].

7.2 Review of ED interpretation systems

Line drawing interpretation, which is an origin of ED interpretation, has been studied since the very beginning of image processing, analysis and machine vision in the late 1960s and early 1970s. Study has been motivated variously by the need to input specific sets of drawings into specific computer-based tools, interest in drawing interpretation as an instance of pattern recognition or as a forcing domain for research in knowledge-based image understanding. It is more than 20 years ago when first papers and systems appeared and were used mainly for map automatic vectorization. Probably, the first bibliography of main papers on this subject was published in 1992 by Kasturi and O'Gorman [42].

After being successfully in image vectorization, researchers moved to recognition of map objects and ED entities. There were many attempts to extract lines, arcs from line-drawing images in 1980s. Then, from the late 1980s, there started to appear papers to recognize more complicated entities like dimensions, circles, dashed lines, etc. In 1990s, researchers started to publish papers describing full systems for ED image interpretation. This field is now beginning to show signs of maturity: a sizeable, stable research community; established, dedicated conferences, workshops and journals; some level of consensus in the techniques and approaches adopted; a growing interest in performance evaluation techniques. We start consideration of this topic from analysis of ED interpretation systems published recently.

A complete system of Kasturi's team for interpretation of mechanical-EDs and building 3D object model from ED projections is described in [26]. It includes algorithms for text/graphics separation, recognition of arrowheads, tails, and witness lines; association of feature control frames and dimension text with the corresponding dimension lines; and detection of dashed lines, sectioning lines, and other objects. A previous version of this team's system has been described in [41].

The system developed by Dori's team for recognition of ED is based orthogonal zig-zag (OZZ) algorithm [30, 35]. The underlying idea of OZZ is inspired by a light beam conducted by an optic fiber: a one-pixel-wide 'ray' travels through a black pixel area designating a bar, as if it were a conducting pipe. Accumulated statistics of the two sets of black run-lengths gathered along the way provide data for deciding about the presence of a bar, its endpoints and its width, and enable-skipping junctions. It is shown that the sparse-pixel approach of OZZ results in about a twenty-fold reduction in both space and time complexity compared to Hough Transform, while the recognition quality is about 40% higher.

A system for recognizing a large class of EDs characterized by alternating instances of symbols and connection lines was developed by Yuhong et al [78]. The output of the system, a netlist identifying the symbol types and interconnections, may be used for design simulation or as a compact portable representation of the drawing. The automatic recognition task is divided into two stages: 1) domain-independent rules are used to segment symbols from connection lines in the drawing image that has been thinned, vectorized, and preprocessed in routine ways; 2) a drawing understanding subsystem works in concert with a set of domain-specific matchers to classify symbols and correct errors automatically. A graphical user interface is provided to correct residual errors interactively and to log data for reporting errors objectively.

Several systems (CELESSTIN, REDRAW) have been developed by Tombre et al. for interpretation of mechanical drawings [68, 14]. The knowledge rules used in the CELESSTIN system are the following: technologically significant entities are extracted and the whole image is analyzed with respect to disassembling and kinematic knowledge. The last version of the CELESSTIN uses knowledge rules relative to the semantics, i.e. to the functionalities of the recognized objects and not only the representation rules. The REDRAW system [1] uses a priori knowledge to achieve interpretation at a semantic level and is based on a model-driven system that can be completely parameterized. A priori knowledge about the domain induces a particular interpretation process for each document class. Among recognized objects, the recognition of parallel lines (hatched areas) is performed.

A complete system to interpret architectural drawings is developed by Tombre's team [37]. The system includes algorithms for symbol and 2D architectural entities recognition as well as 3D modeling process to reconstruct 3D model of a flat. 2D vectorization stage starts from text/graphics segmentation stage. Then, skeletonization and polygonal approximation is performed that includes arc detection as separate step. Dashed lines are detected and symbols are recognized. Finally, 3D modeling is performed to reconstruct floor through elevation. Powerful and flexible interface is also developed.

The ANON system for ED interpretation was developed by Joseph and Pridmore [40] and is based on the combination of schemata describing prototypical drawing with a library of low-level image analysis routines and a set of explicit control rules. The system works directly with raster images without prior thresholding and vectorization, combining the extraction of primitives with their interpretation. Bottom-up and top-down strategies are integrated into a single framework.

A system to convert scanned EDs into a vectorized file has been developed by Chen et al [18]. The vectorized file consists of lines, arcs, and circles and generates a description of paper-based EDs. The scanned ED is initially vectorized by a system called RENDER using line tracking and curve fitting techniques. However, the results obtained after the initial vectorization are not adequate. Then, a post processing system called P-RENDER is described, which has been developed to further refine the vectorized line drawings and to recreate the drawing with the exact numbers of lines and arcs. Dashed entities are detected and recreated. Further, these results have been extended in [47] to extract specific features from ED. Identification algorithm is based on pattern matching process where the entities present in the vectorized drawing are checked for specific pattern primitives.

A system for interpretation of ED images and building 3D object model from vectorized projections is developed by Ablameyko et al [4, 6]. The vectorization process is based on line-by-line processing scheme and run-length image representation [2]. The recognition stage is aimed at obtaining a representation of ED image in terms of universal ED entities: arcs, circles, line types, blocks, crosshatched areas, dimensions and others. The algorithms for recognition of main ED entities were developed. Specifics of the processed ED images expressed in knowledge form are widely used at both vectorization and recognition stages. A step towards 3D object reconstruction has been also successfully made [9].

A system for recognizing technical drawings of Boeing Company (USA) is developed by Robinson et al [59]. It recognizes and uses information within airplane-related vector and raster images. Such images include troubleshooting charts, fault reporting diagrams, component location diagrams, component index tables, wiring diagrams, system schematics, parts illustrations, standards tables, and structural and tooling drawings. Developed tools allow one to automatically convert and integrate electronic data into industry standard formats. Some of the technical challenges include: a) handling a wide variety of source formats, b) making sure that the tools scale up to handle millions of pages of information, and c) adding functionality to graphics. The system contains over four million pages of text including tens of thousands of graphics. The system also explores visual information retrieval strategies, including content-based and similarity-based methods for both vector and raster graphics.

A line network oriented global vectorization system was developed by Song et al [62]. This system uses global information about image to vectorize a line in one step, and carries out the global vectorization of line networks. Therefore, the problem of separating one line is solved, and a complex analysis of crossings is avoided. The performance of vectorization is improved clearly. Furthermore, it can vectorize lines in any orientations well, and can vectorize a dashed line in one step. Aided by the related knowledge, local details of vectorization are refined.

A novel raster-based approach for vectorization process is proposed in [63]. It is called the object-oriented progressive-simplification-based vectorization model. An integrated set of object-oriented pixel-based vectorization algorithms for various classes of graphic objects in engineering drawings based on the idea of progressively simplifying the complexity of raster image during the vectorization process is proposed.

A case-based scheme for engineering drawing recognition is proposed by Yan and Wenyin [76]. The key idea of the scheme is that the user interactively provides an example of one type of graphic object in an engineering drawing, then the system learns the graphical knowledge of this object type from the example and uses this learned knowledge to recognize or search for similar graphic object in engineering drawings.

A vectorization system for architecture EDs is presented in [64]. The system employs the line-symbol-text vectorization workflow to recognize graphic objects in the order of increasing characteristic complexity and progressively simplify the drawing image by removing recognized objects from it. Various recognition algorithms for basic graphic types have been developed and efficient interactive recognition methods are proposed as complements to automatic processing. Based on dimension recognition and analysis, the system reconstructs the literal dimension for vectorization results, which yields optimized vector data for CAD applications.

Analyzing literatures in ED interpretation systems, one can see the following tendencies in task development:

- improving quality of vectorization because this is crucial for further entity recognition stage;
- extending class of recognized entities;
- attracting and formalizing knowledge for entities recognition;
- increasing level of automation due to using more automatic recognition techniques; and
- moving to 3D object reconstruction from vectorized ED projections.

There have also been commercial systems for bitmap vectorization and ED interpretation. Their details are omitted here because the methodologies of those commercial systems are often concealed and comprised of a chunk of heuristics. A list of commercial products can be found in [79].

8 Future tasks

Although the ED analysis field has been intensively developed during the last decades, there are still many unsolved problems. The systems that have been developed are mainly oriented toward interpretation of quite simple drawings. So, in the near future, development of techniques for automatic interpretation of various types of EDs will continue and grow.

We can mention the following main tasks that should be solved in near future:

- robust entity recognition in the presence of other intersecting and overlapping structures;
- separation of touched characters from graphics;
- methods for extraction and representation of consistent 2D scenes from groups of entities;
- robust recognition of symbols, characters, and words with varying fonts, orientations, scale, etc.;
- performance evaluation for recognition of all types of entities;
- efficient editing tools for recognized drawings;
- knowledge representation and use in entity recognition.

Research in the field continues apace and our capabilities in these areas will surely develop in the future. Many prospective solutions to these problems are and will be based on syntactical and hybrid recognition methods. It is clearly important that domain-specific knowledge be applied wherever possible, but this must be done in a principled manner.

Interpretation of EDs with undefined entities (i.e., EDs not formatted in ANSI or other standards) is also a challenging task. In this task, we should firstly find several common line drawings within an ED as the candidates of the entities. We may employ the technique in [16], which finds frequent subgraphs from a large graph representing an entire line drawing.

9 Conclusion

In this paper, we analyzed a problem of ED interpretation. First, we reviewed recognition algorithms of ED entities, while focusing the recognition of dimensions and crosshatching areas. Especially, we analyzed how dimension texts can be separated from graphics and how arrowheads are recognized that are important features in dimension recognition. Second, we also analyzed recent complete or near-complete ED interpretation systems. Finally, future tasks were discussed.

In spite of a quite long period of studying dimension recognition (more than 10 years), the task is far from its full solution. First, only several entity types (restricted set) have been analyzed. Second, robustness and behavior of algorithms especially in bad quality and noisy images is not studied well. Third, performance evaluation of the existed algorithms is not done. That leads us to many questions like for example the following [67]. How to correctly evaluate the result of recognition? What errors can be allowed in the result of recognition? Should errors be grouped according to their type?

Consequently, it is hard to clearly answer the question whether general techniques for entities recognition are actually available, or the state of the art in this field is rather a collection of domain-dependent algorithms.

Among the many algorithms that are in use for entities recognition, which are the best for any ED? Several of them are already well established as regards the obtainable results and could be applied in different contexts. To what extent are they transportable from one drawing type to another?

All these questions lead that the interpretation of ED images will be an active and fascinating area of research and system development for years to come.

References

[1] S. Ablameyko, *An Introduction to Interpretation of Graphic Images*, SPIE Press, USA, TT27, 1997.

[2] S. Ablameyko, V. Bereishik, et al., "System for automatic vectorization and interpretation of graphic images," Pattern Recognition and Image Analysis, vol. 3, no. 1, pp. 39-52, 1993.

[3] S. Ablameyko, V. Bereishik, and P. Foyer, "Recognizing engineering drawing entities: technology and results," Proc. 6th Int. Conf. Image Processing and its Applications, vol. 2, pp.736-740, 1997.

[4] S. Ablameyko, V. Bereishik, O. Frantskevich, E. Mel'nik, M. Khomenko, and N. Paramonova, "Interpretation of engineering drawings: techniques and experimental results," Pattern Recognition and Image Analysis, vol. 5, no. 3, pp. 380-401, 1995.

[5] S. Ablameyko, V. Bereishik, O. Frantskevich, M. Homenko, and N. Paramonova, "Algorithms for recognition of the main engineering drawing entities," Proc. 4th Int. Conf. Document Analysis and Recognition, vol. 2, pp. 776-779, 1997.

[6] S. Ablameyko, V. Bereishik, O. Frantskevich, M. Homenko, and N. Paramonova, "A system for automatic recognition of engineering drawing entities," Proc. 14th Int. Conf. Pattern Recognition, vol. 2, pp. 1157-1159, 1998.

[7] S. Ablameyko, V. Bereishik, O. Frantskevich, M. Homenko, and N. Paramonova, "Knowledge-based recognition of crosshatched areas in engineering drawings, Advances in Pattern Recognition," Proc. Joint IAPR Int. Workshops SSPR'98 and SPR'98, pp. 460-467, 1998.

[8] S. Ablameyko, V. Bereishik, O. Frantskevich, M. Homenko, and N. Paramonova, "Recognition of thick closed polygons in engineering drawing images," Proc. SPIE, vol. 3967, pp. 140-146, 2000.

[9] S. Ablameyko and V. Bereishik, A. Gorelik, and S. Medvedev, "3D object reconstruction from engineering drawing projections," Computing & Control Engineering Journal, vol. 10, no .6, pp. 277-284, 1999.

[10] S. Ablameyko and A. Gorelik, "Parameterisation of automatically vectorized engineering drawings," Proc. 6th Int Conf. Advanced Computer Systems, pp.131-137, 1999.

[11] S. Ablameyko and T. Pridmore, *Machine interpretation of line drawing images*, Springer, 2000.

[12] S. Adam, J. M. Ogier, C. Cariou, R. Mullot, J. Labiche, and J. Gardes, "Symbol and character recognition: application to engineering drawings," Int. J. Document Analysis and Recognition, vol. 3, no. 2, pp. 89-101, 2000.

[13] C. Ah-Soon and K. Tombre, "A step towards reconstruction of 3-D CAD models from engineering drawings," Proc. 3rd Int. Conf. Document Analysis and Recognition, vol. 1, pp. 331-334, 1995.

[14] D. Antoine, S. Colin, and K. Tombre, "Analysis of technical documents; the REDRAW system," Proc. the IAPR Workshop on Structural and Syntactic Pattern Recognition, New Jersey, pp. 192-230, 1990.

[15] J. F. Arias and R. Kasturi, "Efficient extraction of primitives from line drawings composed of horizontal and vertical lines," Machine Vision and Applications, vol. 10, no. 4, pp. 214-221, 1997.

[16] E. Barbu, P. Heroux, S. Adam, and E. Trupin, "Frequent graph discovery: application to line drawing document images," Electronic Letters on Computer Vision and Image Analysis, vol. 5, no. 2, pp. 47-54, 2005.

[17] L. Boatto, et al., "An interpretation system for land register maps," IEEE Computer, vol. 25, no. 7, pp. 25-33, 1992.

[18] Y. Chen, N. Langrana, and A. Das, "Perfecting vectorized mechanical drawings," Computer Vision and Image Understanding, vol. 63, no. 2, pp. 273-286, 1996.

[19] K. -Z. Chen, X. -W. Zhang, Z. -Y. Ou, and X. -A. Feng, "Recognition of digital curves scanned from paper drawings using genetic algorithms," Pattern Recognition, vol. 36, no. 1, pp. 123-130, 2003.

[20] A. K. Chhabra, "Graphic symbol recognition: an overview," Selected Papers from Int. 2nd Workshop on Graphics Recognition Algorithms and Systems, (Lecture Notes in Computer Science, vol. 1389), pp. 68-79, 1998.

[21] A. K. Chhabra and I. T. Phillips, "The second international graphics recognition contest-raster to vector conversion: a report," Selected Papers from Int. 2nd Workshop on Graphics Recognition Algorithms and Systems, (Lecture Notes in Computer Science, vol. 1389), pp. 390-410, 1998.

[22] S. Collin and D. Colnet, "Analysis of dimension in mechanical engineering drawings," Proc. IAPR Workshop on Machine Vision Applications, pp. 105-108, 1990.

[23] S. Collin and P. Vaxiviere, "Recognition and use of dimensioning in digitized industrial drawings," Proc. 1st Int. Conf. Document Analysis and Recognition, pp. 161-169, 1991.

[24] S. Collin and D. Colnet, "Syntactic analysis of technical drawing dimensions," Int. J. Pattern Recognition and Artificial Intelligence, vol. 8, no. 5, pp. 1131-1148, 1994.

[25] L. P. Cordella and M.Vento, "Symbol recognition in documents: a collection of techniques?," Int. J. Document Analysis and Recognition, vol. 3, no. 2, pp. 73-88, 2000.

[26] P. M. Devaux, D. B. Lysak, and R. Kasturi, "A complete system for the intelligent interpretation of engineering drawings," Int. J. Document Analysis and Recognition, vol. 2, no. 2-3, pp. 120-131, 1999.

[27] D. Dori, "A syntactic/geometric approach to recognition of dimensions in engineering machine drawings," Computer Vision, Graphics, and Image Processing, vol. 47, no. 3, pp. 271-291, 1989.

[28] D. Dori, "Syntax-enhanced parameter learning for recognition of dimensions in engineering machine drawings," Int. J. Robotics and Automation, vol. 5, no. 2, pp. 59-67, 1990.

[29] D. Dori, "Dimensioning analysis: toward automatic understanding of engineering drawings," Commun. ACM, vol. 35, no. 10, pp. 92-103, 1992.

[30] D. Dori, et al., "Sparse-pixel recognition of primitives in engineering drawings," Machine Vision and Applications, vol. 6, no. 2-3, pp. 69-82, 1993.

[31] D. Dori and A. Pnueli, "The grammar of dimensions in machine drawings," Computer Vision, Graphics, and Image Processing, vol. 42, no. 1, pp. 1-18, 1988.

[32] D. Dori and M. Weiss, "A scheme for 3D object reconstruction from dimensioned orthographic views," Engineering Applications of Artificial Intelligence, vol. 9, no. 1, pp. 53-649, 1996.

[33] D. Dori and Y. Velkovitch, "Segmentation and recognition of dimensioning text from engineering drawings," Computer Vision and Image Understanding, vol. 69, no. 2, pp. 196-201, 1998.

[34] D. Dori and L. Wenyin, "Vector-based segmentation of text connected to graphics in engineering drawings," Proc. 6th Int. Workshop on Advances in Structural and Syntactical Pattern Recognition, (Lecture Notes in Computer Science, vol. 1121), pp. 322-331, 1996.

[35] D. Dori and L. Wenyin, "Automated CAD conversion with the machine drawing understanding system: concepts, algorithms, and performance," IEEE Trans. Systems, Man and Cybernetics, vol. 29, no. 4, pp. 411-416, 1999.

[36] D. Dori, L. Wenyin and M. Peleg, "How to win a dashed line detection contest," Proc. Int. Workshop on Graphics Recognition, (Lecture Notes in Computer Science, 1072), pp. 286-300, 1996.

[37] P. Dosch, K. Tombre, C. Ah-Soon, and G. Masini, "A complete system for the analysis of architectural drawings," Int. J. Document Analysis and Recognition, vol. 3, no. 2, pp. 102-116, 2000.

[38] D. Elliman, "TIF2VEC, An algorithm for arc segmentation in engineering drawings," Selected Papers from 4th Int. Workshop on Graphics Recognition Algorithms and Applications, (Lecture Notes in Computer Science, vol. 2390), pp. 350-358, 2002.

[39] J. Gao, L. Tang, L. Wenyin, and Z. Tang, "Segmentation and recognition of dimension texts in engineering drawings," Proc.3rd Int. Conf. Document Analysis and Recognition, vol. 1, pp. 528-31, 1995.

[40] S. H. Joseph and T. P. Pridmore, "Knowledge-directed interpretation of mechanical engineering drawings, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 14, no. 9, pp. 928-940, 1992.

[41] R. Kasturi, et al., "A system for interpretation of line drawings," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 10, pp. 978-992, 1990.

[42] R. Kasturi and L. O'Gorman, "Document image analysis - a bibliography," Machine Vision and Applications, vol. 5, no. 3, pp. 231-243, 1992.

[43] T. Kasvand, "Linear textures in line drawings," Proc. 8th Int. Conf. Pattern Recognition, pp. 398-401, 1986.

[44] B. Kong, et al., "A benchmark: performance evaluation of dashed-line detection algorithms," Proc. Int. Workshop on Graphics Recognition, (Lecture Notes in Computer Science, vol. 1072), pp. 270-285, 1996.

[45] C. P. Lai and R. Kasturi, "Recognition of characteristic symbols in engineering drawings," Proc. SPIE, vol. 1661, pp. 83-89, 1992.

[46] C. P. Lai and R. Kasturi, “Detection of dimension sets in engineering drawings,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 16, no. 8, pp. 848-855, 1994.

[47] N. A. Langrana, Y. Chen, and A. K. Das, “Feature identification from vectorized mechanical drawings,” Computer Vision and Image Understanding, vol. 68, no. 2, pp. 127-45, 1997.

[48] C. Lin and C. K. Ting, “A new approach for detection of dimensions set in mechanical drawings,” Pattern Recognition Letters, vol. 18, no. 4, pp. 367-373, 1997.

[49] J. Llados, E. Valveny, G. Sanchez, and E. Marti, “Symbol recognition: current advances and perspectives,” Selected Papers from 4th Int. Workshop on Graphics Recognition Algorithms and Applications, (Lecture Notes in Computer Science, vol. 2390), pp. 104-127, 2002.

[50] Z. Lu, “Detection of text regions from digital engineering drawings,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 4, pp. 431-439, 1998.

[51] H. Luo and I. Dinstein, “Using directional mathematical morphology for separation of character strings from text/graphic images”, Int. Workshop Structural and Syntactic Pattern Recognition, pp. 372-381, 1995.

[52] H. Luo and R. Kasturi, “Improved directional morphological operations for separation of characters from maps/graphics,” Selected Papers from Int. 2nd Workshop on Graphics Recognition Algorithms and Systems, (Lecture Notes in Computer Science, vol. 1389), pp. 35-47, 1998.

[53] S. Meeran, J. M. Taib, and M. T. Afzal, “Recognizing features from engineering drawings without using hidden lines: a framework to link feature recognition and inspection systems,” Int. Journal of Production Research, vol. 41, no. 3, pp. 465-495, 2003.

[54] W. Min, Z. Tang, and L. Tang, “Using web grammar to recognize dimensions in engineering drawings,” Pattern Recognition, vol. 26, no. 9, pp. 1407-1416, 1993.

[55] L. O’Gorman and R. Kasturi, Eds, *Document image analysis*, IEEE Press, 1995.

[56] I. .T. Phillips, J. Liang, and R. Haralick, “A performance evaluation protocol for engineering drawing recognition systems, Proc. 2nd Int. Workshop on Graphics Recognition, pp. 333-346, 1997.

[57] I. T. Phillips and A. K. Chhabra, "Empirical performance evaluation of graphics recognition systems," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 21, no. 9, pp. 849-869, 1999.

[58] G. Priestnall, R. E. Marston, and D. G. Elliman, "Arrowhead recognition during automated data capture," Pattern Recognition Letters, vol. 17, no. 3, pp. 277-286, 1996.

[59] M. L. Robinson, L. S. Baum, J. H. Boose, D. B. Shema, and S. C. Chew, "Case study: Boeing intelligent graphics for airplane operations and maintenance," Proc. XML Europe '99 Conference, pp. 467-478, 1999.

[60] P. L. Rosin and G. W. Wesst, "Segmentation of edges into lines and arcs," Image and Vision Computing, vol. 7, no. 2, pp. 109-114, 1989.

[61] E. Saund, "Finding perceptually closed paths in sketches and drawings," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 25, no. 4, pp. 475-491, 2003.

[62] J. Song, F. Su, J. Chen, S. Cai, "A knowledge-aided line network oriented vectorization method for engineering drawings," Pattern Analysis and Applications, vol. 3, no. 2, pp. 142–152, 2000.

[63] J. Song, F. Su C. -L. Tai, S. Cai, "An object-oriented progressive-simplification-based vectorization system for engineering drawings: model, algorithm, and performance," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 8, pp. 1048-1060, 2002.

[64] F. Su, J. Song, and S. Cai, "A vectorization system for architecture engineering drawings," Proc. 6th IAPR Int. Workshop on Graphics Recognition, (Lecture Notes in Computer Science vol.3926), pp. 11-22, 2005.

[65] F. Su, J. Song, C. -L. Tai, and S. Cai, "Dimension recognition and geometry reconstruction in vectorization of engineering drawings," Proc. Int. Conf. Computer Vision and Pattern Recognition, pp. I-710–I-716, 2001.

[66] K. Tombre, "Graphics recognition-general context and challenges," Pattern Recognit. Lett., vol. 16, no. 9, pp. 883-891, 1995.

[67] K. Tombre, "Analysis of engineering drawings: state of the art and challenges," Selected Papers from Int. 2nd Workshop on Graphics Recognition Algorithms and Systems, (Lecture Notes in Computer Science, vol. 1389), pp. 257-264, 1998.

[68] P. Vaxiere and K. Tombre, "Celesstin: CAD conversion of mechanical drawings," IEEE Computer, vol. 25, no. 7, pp. 46-54, 1992.

[69] L. Wendling, S. Tabbone, "A new way to detect arrows in line drawings," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 26, no. 7, pp. 935-941, 2004.

[70] L. Wenyin, "On-line graphics recognition: state-of-the-art," Graphics Recognition Recent Advances and Perspectives, (Lecture Notes in Computer Science, vol. 3088), pp. 289-302, 2004.

[71] L. Wenyin and D. Dori, "A protocol for performance evaluation of line detection algorithms," Machine Vision and Applications, vol. 9, no. 5-6, pp. 240-250, 1997.

[72] L. Wenyin and D. Dori, "A proposed scheme for performance evaluation of graphics/text separation algorithms," Selected Papers from Int. 2nd Workshop on Graphics Recognition Algorithms and Systems, (Lecture Notes in Computer Science, vol. 1389), pp. 359-371, 1998.

[73] L. Wenyin and D. Dori, "Performance evaluation of graphics recognition algorithms: principles and applications," Proc. 14th Int. Conf. Pattern Recognition, vol. 2, pp. 1180-1182, 1998.

[74] L. Wenyin and D. Dori, "Incremental arc segmentation algorithm and its evaluation," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 4, pp. 424-431, 1998.

[75] L. Wenyin and D. Dori, "From raster to vectors: extracting visual information from line drawings," Pattern Analysis and Applications, vol. 2, no. 1, pp. 10-21, 1999.

[76] L. Yan and L. Wenyin, "Engineering drawings recognition using a case-based approach," Proc. 7th Int. Conf. Document Analysis and Recognition, pp. 190- 194, 2003.

[77] Y. H. Yu, A. Samal, and S. Seth, "Isolating symbols from connection lines in a class of engineering drawings," Pattern recognition, vol. 27, no. 3, pp. 391-404, 1994.

[78] Y. Yuhong, A. Samal, and S. C. Seth, “A system for recognizing a large class of engineering drawings,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 8, pp. 868-890, 1997.

[79] http://www.cvc.uab.es/iapr-tc10/resources/mainResources.htm#links