Symbol Recognition: Current Advances and Perspectives

Josep Lladós, Ernest Valveny, Gemma Sánchez, and Enric Martí

Computer Vision Center, Dept. Informàtica, Universitat Autònoma de Barcelona
08193 Bellaterra (Barcelona), Spain
{josep,ernest,gemma,enric}@cvc.uab.es

Abstract. The recognition of symbols in graphic documents is an intensive research activity in the community of pattern recognition and document analysis. A key issue in the interpretation of maps, engineering drawings, diagrams, etc. is the recognition of domain dependent symbols according to a symbol database. In this work we first review the most outstanding symbol recognition methods from two different points of view: application domains and pattern recognition methods. In the second part of the paper, open and unaddressed problems involved in symbol recognition are described, analyzing their current state of art and discussing future research challenges. Thus, issues such as symbol representation, matching, segmentation, learning, scalability of recognition methods and performance evaluation are addressed in this work. Finally, we discuss the perspectives of symbol recognition concerning to new paradigms such as user interfaces in handheld computers or document database and WWW indexing by graphical content.

1 Introduction

Symbol recognition is one of the significant applications within the area of pattern recognition. Fields like architecture, cartography, electronics, engineering etc. use domain-dependent graphic notations to develop their designs. The automatic interpretation of such documents, requires processes able to recognize the corresponding alphabets of symbols. Because of this wide range of graphic documents, each one with its own characteristic set of symbols, it is not easy to find a precise definition of a symbol. In a very general way, a symbol can be defined as a graphical entity with a particular meaning in the context of an specific application domain. Thus, and depending on the application, we can find different kinds of symbols according to their visual properties: simple 2D binary shapes composed of line segments (engineering, electronics, utility maps, architecture), a combination of line segments and solid shapes (musical scores), complex gray level or color shapes (logos), silhouettes (geographic symbols), etc. In this paper, we have taken the general definition stated above and hence, we have focused our attention on applications and methods developed to identify any meaningful entity in graphic documents. Good reviews on the state-of-the-art about symbol recognition were reported in previous Graphics Recognition Workshops and

D. Blostein and Y.-B. Kwon (Eds.): GREC 2002, LNCS 2390, pp. 104–128, 2002.

related conferences [1, 2, 3, 4]. The main goals of this new overview are: first, to update the literature review on symbol recognition; secondly, to give a systematic and structured overview of methods, providing a double and related classification from two points of view, namely, applications and techniques. Finally, to address the set of challenges and open issues which can be derived from the analysis of current approaches.

From the point of view of applications, much of the research in graphics recognition has been addressed to the automatic conversion of graphic documents to a format able to be understood by CAD systems. Many efforts have been focused on the development of efficient raster-to-vector converters (e.g. [5, 6, 7, 8]). Moreover, performance evaluation methods have been developed [9] and contests on raster-to-vector conversion have been held in past editions of the Graphics Recognition workshop [10, 11]. However, raster-to-vector conversion should not be the final step. Complete raster-to-CAD conversion should also provide a semantic description and interpretation of the drawing. In this context, symbol recognition is required to identify graphic entities and it has been applied to many applications, including interpretation of logic circuit diagrams, engineering and architectural drawings and any kind of maps. Apart form raster-to-CAD conversion, other significant applications where symbol recognition plays an important role are interpretation of musical scores for conversion to MIDI and logo recognition. In section 2, we will review these application domains, pointing out the most relevant properties of each domain for symbol recognition. From the point of view of methods, symbol recognition is a particular application of pattern recognition. In section 3 we will describe symbol recognition according tothe classical classification in statistical and structural approaches. In table 1, we summarize symbol recognition literature according to this double classification: application domains and methods for recognition.

From the analysis of all these methods, we can conclude that although many different approaches have been reported, it is difficult to find a general and robust method covering different domains with a well-established and evaluated performance. Authors tend to develop ad-hoc techniques which are difficult to be reused in other domains. Moreover, there is no way to compare different methods operating in the same domain or to validate the performance of any approach with a significant set of symbols and drawings. In section 4, we address and identify some open problems involved in the development of general, robust and efficient symbol recognition methods. We draw conclusions from the current state of art and outline some challenges for future research. Among these issues we have included symbol segmentation and representation, matching, learning, scalability and performance evaluation.

In the last section we discuss future perspectives for symbol recognition. New paradigms in the information society technology, such as keyboardless user interfaces for handheld computers, internet search engines based on graphical queries, on-line symbol recognition in graphics tablets and interactive pen displays, etc. also demand for symbol recognition capabilities, but with different requirements than classic applications.

Table 1. State of the art of symbol recognition in a twofold point of view: application domains and techniques

	logic diagrams	engineering drawings	maps	musical scores	architectural drawings	logo recognition	formula recognition	other applications
Structural matching	[12]–[16]	[17, 18]			[19, 20]	[21, 22]		[23]
Syntactic approaches	[24]–[26]	[27]–[30]		[31]	[32]		[33, 34]	[24]
constraint satisfaction	[35, 36]	[37]	[38, 39]		[40, 41]			[42]
Neural networks	[43]		[44]	[45]–[48]		[49, 50]		
Statistical classifiers	[51, 52]		[53]–[58]	[59]	[41]	[60]–[62]	[63]	[64]
decision trees	[65]							[66, 67]
Heuristic/ad-hoc techniques		[68]	[69, 55, 70]	[71]			[72]	[73]
Other	[13]	[74]						[75]

2 Application Domains of Symbol Recognition

2.1 Logic Circuit Diagrams

Electrical and logic circuit diagrams is one of the earliest application domains that focused its attention on graphical symbol recognition. A lot of contributions can be found in the literature [12, 36, 13, 51, 26, 14, 15]. The understanding and validation of electrical schematics and its conversion to an electronic format has become through the years a prototypical graphics recognition application. Circuit diagrams offer two advantages that have probably contributed to that. First, they have a standardized notation which is based on loop structures that characterize the symbols, and rectilinear connections between them. Such representational dichotomy between symbols and interconnections leads to the second advantage: symbols belonging to logic diagrams can be segmented in a reasonably easy way by distinguishing between lines and loops or background areas, in addition to small connected components, likely representing textual annotations.

2.2 Engineering Drawings

The first difficulty of engineering drawings is that we can not assume a standardized diagrammatic notation. Actually we can distinguish two levels of symbols. The first level consists of graphical entities that can have a different meaning depending on the context where they appear. Ablameyko [76] distinguished four types of graphical entities: arcs and straight lines, dashed lines, crosshatched areas, and dimensions. These primitives can represent an angular information, a section of a mechanical part, a symmetry axis, etc. Symbols at the second level are formed by an assembly of the low level primitives. The recognition of these elements combined with domain-dependent knowledge, gives meaning to the document and allows it to be converted to a GIS or CAD format. The problem of arc detection was recently studied in [77] proposing a method that combines two of the most reliable techniques in the literature. Hatched pattern detection is an important concern in the field of document analysis [78, 69, 73] and is usually solved by clustering parallel lines having the same slope angle and sorting them along a normal direction. Dimensions usually follow strict standards. Their interpretation and validation is very important, not only to fully understand the document but also to assist in the segmentation of other graphical entities. Since they are usually based in combinations of arrowheads, lines and text in particular configurations, syntactic approaches, usually based in graph grammars, are the most usually employed techniques [27, 28, 30]. Concerning to higher level symbols that usually represent mechanical parts, since they are very domain-dependent and even document-dependent, the automatic recognition cannot be made fully automatic and requires special interactive techniques and knowledge.

2.3 Maps

The conversion of maps to a GIS format has several challenges as the combination between cartographic information and satellite images, or the integration and conversion of maps from different areas (cadastral, telephone, water, etc.). From a general point of view, we could define three types of maps which have their own notational conventions. At the lowest level of difficulty, we could place cadastral city maps [69, 55, 79, 70]. In such type of documents, symbols have a polygonal shape often filled by a hatching pattern. These polygonal shapes represent parcels and the surrounding streets and their meaning is completed with text and annotations. Thus, symbol recognition is usually formulated in terms of detection of polygonal shapes and hatched patterns. Another subdomain of map interpretation focus on utility maps [53, 54, 39]. Utility maps contain information on network facilities of companies such as water, telephone, gas, etc. They are usually binary images consisting of lines and small symbols composed of geometric basic primitives (squares, circles, arrowheads, etc.). The recognition is usually very domain dependent. Finally, geographic maps [58, 38, 44, 57] are probably the most difficult class of documents of this domain. In this kind of maps, the graphical entities are associated with line objects that usually represent isolines and roads and, on the other hand, small symbols whose meaning is given by a legend. Color information plays an important role. Thus, a layer segmentation process is usually performed in terms of color quantization. Line objects are characterized by regular structures that can be represented by a linear grammar, i.e. they are composed of regular combinations of lines, points and gaps. Thus, to extract roads and isolines, lines are followed under a rule-based criterion. The detection of symbols usually is legend-driven, i.e. the legend is first detected and then, symbol patterns are extracted from it. The meaning of each prototype symbol is also captured by an OCR procedure. Variations of pattern matching are the most used techniques to recognize such symbols.

2.4 Musical Scores

The recognition of musical scores [45, 59, 80, 46, 71, 47] can be considered a graphics recognition domain not only from the point of view of the application but also from the point of view of the proposed procedural solutions. The particular structure of a musical score and its standardized notation have resulted in the development of a set of very specific techniques, only applicable to this family of documents. The interpretation process is organized in three stages. First, the extraction of staff lines, that allows to segment individual symbols, and can be performed by projection or run analysis techniques. Second, the recognition of individual notes that, since there is a finite set of standard symbols, are robustly recognized by neural networks or feature vector distances. The third stage is the interpretation of the whole musical score. The great part of the literature solve this task by using different variations of graph grammars.

2.5 Architectural Drawings

The interpretation of architectural plans is one of the most recent activities [40, 41, 19, 20]. In architectural drawings, the recognition of higher level entities such as walls, doors, windows, furniture, stairs, etc. allows the interpretation of the document and, hence, its conversion to a CAD environment to perform actions as design edition, validation, 3D visualization and virtual navigation inside the building. Two major symbol structures can be categorized: prototype-based symbols and texture-based symbols. Examples of symbols characterized by a prototype are doors and windows. On the other hand, symbols characterized by a structured texture as hatching or tiling represent walls, floors or stairs. Two problems make the recognition of architectural symbols difficult. First, there is no standardized notation and hence, a general framework for the interpretation of the documents is not a solved issue. Second, since symbols appear embedded in the document, its segmentation is difficult to be separated from the recognition. Due to that, recognition has to be done by searching throughout all the document and, hence it is an expensive process.

2.6 Logo Recognition

Logo and trademark recognition [49, 21, 22, 60, 50, 61] can be considered a symbol recognition application that differ from the other categories. While the purpose in the other subdomains is the interpretation of a certain diagram in which symbols are constituent graphical entities following a particular notation, logo recognition is devoted to clustering documents in terms of the originating institution and retrieval by content from document databases. Thus, while classical symbol recognition methods assume that the set of symbols to be recognized in a particular application are "similar" in terms of the constituent features, the recognition of logos requires a more general framework. Due to the unrestricted variety of instances, logo recognition is usually based on extracting signatures from the image in terms of contour codification, invariant moments, connected component labeling, etc. and match the unknown logo with the database models using different types of distance or neural networks. Since logos often combine text and graphics, the recognition in some cases also includes OCR processes.

2.7 Other Applications

In addition to a number of symbol recognition works that has been performed on other types of flow charts and diagrams [24, 73, 64] let us briefly describe three particular applications, namely formula recognition, on-line symbol recognition for pen-based user interfaces and WWW graphic indexing and querying. Mathematical formula recognition is at the frontier between OCR and symbol recognition. Actually, symbols in mathematical formulas can be considered as belonging to a particular font of characters. However, from the point of view of the structure of the formula and its interpretation, the problem falls out the classical OCR approaches. The existing literature [34, 63, 72] uses feature vectors to recognize individual symbols and syntactic approaches to validate the structure of the formula. Symbol recognition procedures are also used as a tool for man-machine interfaces [66, 81, 75, 82, 83]. This is not only an specific application domain but it also requires specific techniques because the recognition is performed on-line. The general goal is to use symbolic shortcuts in a pen-based environment that allow the user to perform operations such as select, delete, copy, or interactively draw in graphical design applications. Finally, a recent application area in which symbol recognition may play an important role is indexing by content on WWW documents. The number of WWW documents in Internet is growing very fast and the ability to make queries by graphical content would allow a more efficient search of information into the Web site. In the last years the problem of locating text in Web images has been addressed [84]. The definition of new XML-based vectorial formats as SVG makes symbol recognition techniques as useful tools to implement search engines based on graphical content.

3 Symbol Recognition Methods

Symbol recognition is one of the multiple fields of application of pattern recognition, where an unknown input pattern (i.e. input image) is classified as belonging to one of the predefined classes (i.e. predefined symbols) in a particular domain. We will take the traditional classification of pattern recognition into statistical and structural approaches to give a systematic and structured overview of symbol recognition methods. The goal is only to describe which methods have been used in symbol recognition, relating them to general pattern recognition strategies. For a more general and detailed discussion of pattern recognition, many excellent surveys and books can be found in the literature [85, 86, 87, 88].

3.1 Statistical Symbol Recognition

In statistical pattern recognition, each pattern is represented as an n-dimensional feature vector extracted from the image. Classification is carried out by partitioning the feature space into different classes, one for each symbol. Therefore, two issues are especially relevant for the performance of this kind of methods:

the selection of the features and the selection of the method for partitioning the feature space. In table 2, we have classified symbol recognition approaches according to these two issues.

The selection of the feature space depends on the properties of the patterns to be classified. The main criterion must be to minimize the distance among patterns belonging to the same class and to maximize the distance among patterns belonging to different classes. Additional interesting properties of feature space are invariance to affine transformations and robustness to noise and distortion. An interesting survey of feature extraction methods, applied to the related area of character recognition can be found in [89]. In symbol recognition, only a subset of all these features have been employed. We will classify them into four groups: those based on the pixels of the image, geometric features, geometric moments and image transformations.

The simplest feature space is the image space itself. The feature vector is composed of one feature for each pixel value. Usually, the image is first normalized to a fixed size. The main advantages are simplicity, low complexity and direct correspondence with visual appearance. However, the representation is not rotation invariant and it is very sensitive to noise and distortion. Another set of methods use geometric features: centroids, axes of inertia, circularity, area, line intersections, holes, projection profiles, etc. In relation with image space, the size of the feature vector can be reduced. A good selection of relevant features is critical to achieve high discrimination power, invariance to affine transformations. Feature extraction must be robust enough to reduce feature variability due to noise and distortion. Moment invariants are another kind of features which have also been applied to symbol recognition, Both the regular moments [51, 43] and the moments of Zernike [53] have been used. Moment invariants are easy to compute, they have relation with geometric properties, such as the center of gravity, the axes of inertia, etc, and they can be made invariant to affine transformations. Finally, features can also be defined through the application of some kind of transformation of the image. Features are taken from the representation of the image in the transformation space. Image transforms which have been used in symbol recognition include Fourier transform [51, 67], Fourier-Mellin transform [53] or special transforms to get signatures from the image [60].

Once a set of features have been chosen, classification consists in selecting a method to partition the feature space and to assign each feature vector to one of the predefined classes. In symbol recognition literature, we can find methods based on the concept of similarity, on neural networks and on decision trees. The

Table 2. Statistical symbol recognition approaches: crossing of features and classification methods

	Image	Geometric features	Moments	Image transformations
Distance-based	[58, 62, 64]	[54, 59, 90, 63, 91, 52, 61]	[51]	[60, 51]
Nearest neighbors		[57]	[53]	[53]
Decision trees		[66, 65]		[67]
Neural networks	[45, 49, 44, 47, 48]	[50, 46]	[43]	

simplest way to partition the feature space consists in defining a distance function among feature vectors and assigning each input image to the class with the closest representative. A slight variation is the *k-nearest neighbors rule*, where several representatives are taken for each class and, for each input pattern, the set of the k closest representatives is built. The pattern is assigned to the class having more representatives in this set. Neural networks have showed to have good classification rates in many different domains. One of their advantages is their learning ability to adapt themselves to the properties of the training set. Learning is performed automatically, providing the optimal parameters of the network to recognize the symbols in the training set. In decision trees, each node of the tree corresponds to an specific condition about the value of a particular feature. Classification is carried out by following the branches in the tree according to the result of condition testing, until one of the leaves is reached. The leaves of the tree correspond to recognized symbols.

3.2 Structural Symbol Recognition

In structural pattern recognition symbols are represented with a description of their shape using some suitable set of geometric primitives and relationships among them. For each symbol, a model of its ideal shape is built using these primitives. An input image is classified as belonging to the symbol giving the best matching between the representation of the image and the model of the symbol. Usually, straight lines and arcs are the primitives used to describe the shape of the symbols although sometimes, other geometric primitives, such as loops, contours or simple shapes (circles, rectangles, etc.) have also been used. Therefore, a previous vectorization step is required. Vectorization can introduce noise and distortion in the representation of images and thus many times, error-tolerant matching must be used.

A large class of structural approaches are based on a graph representation of the symbols [12, 36, 13, 14, 15, 92, 17, 23]. Nodes and edges of the graph correspond, respectively, to points and lines of the image, providing a very natural and intuitive description of symbols. Matching consists in finding the best subgraph isomorphism between the input image and the models of the symbols. With this approach, symbols can be found as subgraphs of the whole image allowing to perform segmentation and recognition at the same time. Distortion is handled using error-tolerant subgraph isomorphism graph edit operations to define an error model. The main drawback of graph matching is computational complexity. Some ways of reducing computation time have been explored.

Formal grammars - usually graph grammars because of bidimensional structure of symbols - are used in another family of structural approaches [24, 80, 25, 26, 34]. A grammar stores in a compact way all valid instances of a symbol or a class of symbols. The recognition of an input image consists in parsing its representation to test if it can be generated by the grammar. To handle distortion, differnt types of error-correcting parsers are proposed. Grammars are useful in applications where the shape of the symbols can be accurately defined by a set of rules, for instance, the recognition of dimension symbols in technical drawings [27, 28, 30] and the recognition of symbols composed of textured areas [32]. Joseph and Pridmore [29] show an alternative use of grammars in which the grammar not only describes the structure of the symbols, but also guides the interpretation of the whole drawing.

Another set of methods uses a set of rules to define geometric constraints among the primitives composing the symbol. Then, these rules are applied or propagated to find symbols in the input image [35, 38, 37]. In [42] a kernel based on a *blackboard architecture* guides the application of the rules, selecting and activating a set of procedures for searching new elements in the drawing when any primitive or symbol is recognized. A similar approach is used in [39]. In [40], the rules are organized in a constraint network. Symbols are identified by traversing the network and testing the rules at every node.

Some other approaches use deformable template matching and a structural representation of the symbols [93, 20] to handle distortion in hand-drawn symbols. The goal is to find a deformation of the model of the symbol resembling the input image. This goal is achieved through the minimization of an energy function, composed of an internal energy measuring the degree of deformation of the model and an external energy measuring the degree of similarity between the deformation of the model and the input image.

Hidden Markov Models can also be seen as structural methods since the structure of the symbol can be described by the sequence of states generating the image. Recognition consists in finding the sequence of states with higher probability. Features used to represent the symbols in HMM approaches include discrete cosine transformation [18], log-polar mapping [21] and image pixels [33]. HMMs are able to segment the symbols and to recognize distorted symbols.

Finally, there is a set of methods [69, 55, 73, 70, 72, 71, 68, 37], also based on a structural representation of the image, in which symbol representation and symbol recognition are not independent tasks. Symbol recognition is carried out by a set of specific procedures for each symbol, and the knowledge about the shape of the symbol is encoded in the procedure itself.

4 Open Issues

From the analysis of existing approaches, we can identify some unsolved or unaddressed issues, concerning the development of general, robust and efficient symbol recognition strategies. In this section, we will discuss the most significant of them, outlining open questions and future perspectives and challenges.

4.1 Segmentation

Symbol segmentation is a very domain-dependent task. A number of symbol recognition contributions assume that symbols have been previously segmented (see appendix A). However, it is not always feasible to break the drawing up into unique constituent components. In certain cases, only a partial or approximate segmentation can be done and domain-dependent knowledge or user assistance is required. The methods that separate segmentation and recognition in different stages, usually base the segmentation on features such as connected components, loops, color layers, long lines, etc. As we have seen in section 2, there are some domains were symbols can be separated from the rest of the drawing in terms of the knowledge of the notation. Thus, the easiest domain is probably musical scores where symbols can be segmented in terms of connected components after removing the staff lines by projections or run length smearing. Other applications such as maps support the segmentation on the presence of a legend. Thus, symbols in the legend are first segmented and then, this information is used as signature to index in the whole document. In electronic diagrams, symbol segmentation is achieved by removing long lines that connect loop-based graphical entities. In other domains such as logo or formula recognition, symbols can be segmented in terms of connected components. Difficulties arise when symbols appear embedded in the drawing. This is usual where symbols consist of an assembly of low level primitives such as arcs, lines, loops and crosshatched or solid areas. Efficient techniques have been proposed to detect each type of these low level primitives. However, at a higher level, considering that a symbol is an assembly of the above primitives, to find a part of the diagram that is likely to represent a symbol is not a trivial task. In such cases, assuming that the knowledge about the domain is required, the trend is to define symbol signatures based on simple features that allow to locate image areas with high evidence to contain a symbol. Doermann in [94] reviews different techniques for indexing of document images that can be used for symbol segmentation.

4.2 Symbol Representation

The selection of an structure for symbol representation can have strong influence in the performance of symbol recognition. It also has a strong relationship with the selection of a matching method, although both issues should be clearly distinguished. A suitable structure should be compact, complete, i.e. general enough to represent symbols from different domains, discriminant, i.e. able to maximize the intra-class similarity and the inter-class dissimilarity, computationally manageable, extensible and able to support distortion models.

In statistical symbol recognition, feature vectors are a very simple representation with low computational cost. However, discrimination power and robustness to distortion strongly depends on the selection of an optimal set of features for each specific application. There is no comparative study on the performance of symbol recognition with different sets of features. Moreover, the number of features must be small and sometimes methods for reduction of dimensionality must be applied. Finally, these methods need a previous segmentation step, which it is not always an easy task because of the embedding of symbols in the drawing.

In structural representations, feature selection is not so critical because they usually rely on a vectorial representation, although vectorization introduces some errors in the representation. The main advantages of structural representations are generality, extensibility and ability to include distortion models.

In conclusion, there is no optimal and general structure for symbol representation and it seems not easy to define a representation powerful enough and general enough to perform well in different domains and applications. Probably, a comparative study on the performance of different representations on several problems and applications is required. Moreover, further research can be done to explore the feasibility of mixed representations combining both approaches and being able to represent symbols in a more general and complete way. This approach can allow the application and combination of several classifiers - see section 4.3 - to get better performance. Signatures have emerged in pattern recognition applications as a simple, flexible and general structure to represent relevant shape properties. Their application to symbol representation could be an interesting approach for general segmentation approaches and applications such as indexing and retrieval of graphic documents.

4.3 Matching

Matching is the procedure to decide to which symbol corresponds an unknown input image. It is the core of any symbol recognition approach. Some desirable properties for matching are the following:

– **Generality:** the ability of matching to be applied to a wide number of different applications.
– **Extensibility:** the ability of matching to work if we add new symbols to be recognized, without need for being changed or rewritten.
– **Scalability:** it refers to the performance of matching when the number of symbols is significantly high (hundreds or thousands), in terms of recognition rates and computation time.
– **Robustness to distortion:** matching must be able to recognize distortions due to noise, hand-drawing or feature extraction errors, without increasing confusions among different symbols.
– **Low computational complexity**.

Concerning to these properties, in statistical methods, generality, scalability and robustness to distortion will depend on the selection of an optimal set of features, although there are no evaluation studies on these subjects. Adding new symbols can require re-learning classification parameters since the feature space is modified. They are usually computationally efficient methods. On the other hand, structural approaches are easier to generalize to different domains and to extend with new symbols to recognize because all graphic symbols share a similar representation using lines and points, the usual low-level primitives. Errors produced by vectorization can increase confusions among different symbols, making matching difficult to scale to a great number of symbols. Most of these methods are able to recognize distorted symbols with different degrees of distortion allowed. Finally, they are methods having high computational complexity, increasing it significantly with the number of symbols to be recognized.

As we can see, many approaches using both statistical and structural approaches have already been developed, and both strategies have attractive properties for symbol recognition. Therefore, maybe research efforts should be concentrated, not in finding new general methods, but in combining several existing methods following some kind of combination scheme. This strategy could lead to significant improvements on performance and could provide new perspectives to the scalability problem. Another interesting issue is that of parallelization of algorithms to reduce computational complexity in structural approaches.

4.4 Learning

Symbol recognition requires the selection of some representative for each symbol type. When patterns are represented by numeric feature vectors, the representative is easy to be computed by the mean or the median of the feature vectors. However, when objects are represented in terms of symbolic structures, the inference of a representative is no longer clear and it is not an obvious task. The computation of a mean symbol is useful in applications that require the learning of a prototypical pattern from a set of noisy samples. Two categories for symbol representative computation can be stated [95]. The fist approach looks for a representative among the set of samples. Thus, given a set of symbols $S = \{s_1, \ldots, s_n\}$, the *set median symbol* $\hat{s}$ is defined as a symbol $s_i \in S$ that has the minimum combined distance to all other elements in the set. Formally,

$$\hat{s} = arg \min_{s_i \in S} \sum_{s_j \in S} d(s_i, s_j), \tag{1}$$

where d denotes the distance function between symbols. In general, the inference of a representative is not constrained to the set of samples but has to be searched among the set of all possible symbols of the particular context. Formally, let $S = \{s_1, \ldots, s_n\}$ be a set of symbols and let Ω be the alphabet of all possible symbols, the *generalized median symbol* or just the *mean symbol* $\bar{s}$ is defined as:

$$\bar{s} = arg \min_{s \in \Omega} \sum_{s_i \in S} d(s, s_i). \tag{2}$$

The theoretical basis for symbol learning can be found in conceptual clustering approaches to learn shapes from examples. Wong and You [96] proposed a statistical-structural combined approach. They defined random graphs as a particular type of graphs which convey a probabilistic description of the data. A process is defined to infer a synthesized random graph minimizing an entropy measure. Segen [23] developed a graph-based learning method for non rigid planar objects able to learn a set of relations between primitives that characterize the shapes of the training set. Recently, Jiang et al. [13] proposed a genetic algorithm to compute the mean among a set of graphs. The algorithm was applied to graphs representing graphical symbols of electronic diagrams. The experimental results prove that the algorithm is able to obtain a representative symbol that smoothes the individual distortions in the noisy samples. Cordella et al. [97] describe the graph representing the symbols with a set of logic predicates. Then, they apply Inductive Logic Programming to modify these predicates from a set of examples. Another recent approach based on deformable models is described in [98], where for each test image, the deformation of the symbol that best fits the image is found. Then, the representative of the symbol is defined as the mean of all deformations. In conclusion, there is still room for improvements in symbol learning. Although there are plenty of contributions in the literature on the symbolic learning paradigm, a few of them experiment their proposed algorithms on graphical symbols frameworks, using real data with noise and distortion as source samples. On the other hand, learning from sets of complex symbols and improving the computational cost are still challenges on that issue.

4.5 Scalability

When the number of symbols to be recognized increases, the recognition performance can degrade because the uncertainty of the inter-class boundaries increases, and hence the probability of symbol confusion. From the point of view of computation time, if recognition requires sequential matching between the input symbol and each model symbol in the database, the time will clearly grow with the number of symbols. Very few symbol recognition methods take into account the scalability issue. Current methods work reasonably well with small databases but their performance seriously degrades when databases contain hundreds of prototypes. Since real applications use to have large databases of symbols, the scalability issue is mandatory to be addressed. It requires the development of new approaches able, firstly, to represent large databases of prototypes and, secondly, to recognize unknown symbols with invariance to the size of the database. One way to perform the search is to use an indexing procedure. The basic idea of indexing is to use a specific set of easily computable features of an input symbol in order to “rapidly” extract from the database those symbols containing this group of features. Some solutions have been proposed in restricted domains. The most outstanding contributions are based on graph representations [99, 100]. These approaches represent symbols by graphs and use variations of the graph adjacency matrix as indexing keys. The database is often organized in terms of a network in which symbols are hierarchically represented taking advantage of their common substructures. Although the matching cost is independent from the size of the database, these approaches are generally restricted to noise free environments and require an exponential time to compile the set of prototypes for the database. Another interesting approach, applied to logo recognition [60], is that of detecting some relevant features of the symbol and matching the input image only with those prototypes in the database sharing those features.

4.6 Performance Evaluation

Performance evaluation is an emerging interest of graphics recognition community. As we have seen in past sections, a lot of methods for symbol recognition have been designed. However, we have already noted the need for evaluation studies to estimate the accuracy, robustness and performance of different methods in some systematic and standard way. Usually, the algorithms are only evaluated by the subjective criteria of their own developers, and based on qualitative evaluation reported by human perception. In the last years, the Graphics Recognition community has reported interesting contributions on performance evaluation of vectorization systems, e.g. [101, 9] and several vectorization contests have been held in past editions of Workshop on Graphics Recognition [10, 11]. These contests have been directed towards the evaluation of raster-to-vector systems, designing metrics to measure the precision of vectorial representations. However, these metrics are not able to evaluate the impact of vectorization on higher-level processes such as symbol recognition. It could be interesting to extend them to be able to evaluate the influence of a vectorization method in symbol recognition.

Liu an Dori [102] distinguished three components to evaluate the performance of graphics recognition algorithms. First, the design of a sound ground truth covering a wide range of possible cases and degrees of distortion. Secondly, a matching method to compare the ground truth with the results of the algorithm. Finally, the formulation of a metric to measure the "goodness" of the algorithm. Following these criteria, a first contest in binary symbol recognition has been organized in the last *International Conference on Pattern Recognition* [103]. The dataset consisted of 25 electrical symbols with small rotation angles, scaled at different sizes and with three types of noise: quantization error, replacement noise and salt-and-pepper noise. Performance measures include misdetection, false alarm and precision of location and scale detection.

It is not an easy task to define a ground-truth and quantitative, domain independent evaluation measures for symbol recognition. The discussion is open, although we will outline some of the guidelines to develop general datasets and metrics. Datasets should be designed in order to include a high number of symbols from different classes and application domains to evaluate the generality and scalability of methods; they should also contain symbols with different kinds and degrees of distortion; non-segmented instances of the symbols are also required to test segmentation ability; finally, real drawings should also be provided to evaluate the performance in real applications. An important issue in designing a dataset is the proper organization and classification of the data. From the point of view of metrics, indices should be developed to measure issues such as recognition accuracy, computation time, degradation of performance with increasing degrees of distortion, degradation of performance with increasing number of symbols, generality, etc.

5 Conclusions and Perspectives

In this paper we have reviewed the state of the art on the symbol recognition problem. It is an update of previous reviews reported by Blostein [1], Chabbra [2] and Cordella [3] in the previous Graphics Recognition Workshops. In the first part of the paper, we have reviewed the literature from two points of view, namely the application domain and the techniques used for the recognition. Afterwards, we have discussed the challenges that the scientific community has in the following years in relation with symbol recognition and related issues.

Many symbol recognition techniques are available, but it is difficult to see a dominant one. The influence of the domain knowledge and the diagrammatic notation properties makes each family of applications to develop its own methods. The definition of a generic symbol recognition method is still a challenge. Several approaches have been proposed in terms of classical pattern recognition methods, either statistical or structural, but they tend to concentrate on a restricted range of requirements. There are other open issues beyond the matching itself. The effort should be made in the development of symbol recognition methods able to combine different classifiers that use different types of constituent features. Therefore, it is also desirable to find a representational model for symbols robust, manageable and general enough to uphold methods from different paradigms and also, to represent symbols from different domains. Symbol segmentation is feasible in those domains where the notation gives enough information about the differences between a symbol and the other graphical entities. However, when symbols appear embedded in the document segmentation and recognition are hardly separated. One solution is the definition of symbol signatures that allow to index into the drawing to locate areas where the symbol is likely to appear. Other issues have been outlined. Thus, the symbol prototypes should be learned, minimizing the intra-class distance and maximizing the inter-class distance. It is a straightforward task when symbols are represented by feature vectors, but becomes a non trivial issue when symbols are represented by structural models. Most symbol recognition methods are not scalable and are just tested with databases of a few number of prototypes, but real applications use to manage sets of hundreds of prototypes. Since the recognition performance uses to degrade with large databases, the robustness against scalability is strongly required to be studied by the research community. Finally, an open issue is the need for protocols to evaluate the performance of symbol recognition algorithms.

In addition to the technological and scientific challenges in the symbol recognition field, the perspectives should also be stated in terms of the potential applications that can take advantage of this technology. Classically, symbol recognition has been integrated in processes involving a paper-to-electronic format conversion. Although there is still room for improvements in the interpretation of paper-based diagrams, the background in symbol recognition can be perfectly used to solve new challenges associated with the evolution of new technologies. One of the promising applications where symbol recognition is a key procedure is document image retrieval by graphical content. There is a wide variety of applications: making queries in document databases to retrieve documents of a given company using its trademark, or indexing in a map database in terms of given symbols or those in the legend. But the same ideas can be applied in Web search engines. Up to now, only textual queries can be made, however extending it to search for graphical entities would be a great help in the navigability. Even in browsing large graphical documents, symbol recognition is a very interesting tool if the user was able to rapidly and interactively retrieve those graphical entities in the document similar with a selected one. The world of pen-based computers and, particularly, PDAs offers also great chances, in that case, taking advantage of the comprehensive work on on-line symbol recognition. Notice that we have stated new perspectives for symbol recognition but which can be supported on the classical techniques. Hence, in addition to investigate in the challenges discussed in section 4, the forthcoming activity within the graphics recognition community will also be concerned in exploring such new perspectives.

References

[1] Blostein, D.: General diagram-recognition methodologies. In Kasturi, R., Tombre, K., eds.: Graphics Recognition: Methods and Applications. Springer, Berlin (1996) 106–122 Vol. 1072 of LNCS. 105, 117

[2] Chhabra, A.: Graphic symbol recognition: An overview. In Tombre, K., Chhabra, A., eds.: Graphics Recognition: Algorithms and Systems. Springer, Berlin (1998) 68–79 Vol. 1389 of LNCS. 105, 117

[3] Cordella, L., Vento, M.: Symbol and shape recognition. In Chhabra, A., Dori, D., eds.: Graphics Recognition: Recent Advances. Springer-Verlag, Berlin (2000) 167–182 Vol. 1941 of LNCS. 105, 117

[4] Kasturi, R., Luo, H.: Research advances in graphics recognition: An update. In Murshed, N., Bortolozzi, F., eds.: Advances in Document Image Analysis, First Brazilian Symposium, BSDIA'97. Springer, Berlin (1997) 99–110 Vol. 1339 of LNCS. 105

[5] Chen, Y., Langrana, N., Das, A.: Perfecting vectorized mechanical drawings. Computer Vision and Image Understanding **63** (1996) 273–286. 105

[6] Dori, D., Liu, W.: Sparse pixel vectorization: An algorithm and its performance evaluation. IEEE Trans. on PAMI **21** (1999) 202–215. 105

[7] Nagasamy, V., Langrana, N.: Engineering drawing processing and vectorisation system. Computer Vision, Graphics and Image Processing **49** (1990) 379–397. 105

[8] Tombre, K., Ah-Soon, C., Dosch, P., Masini, G., Tabonne, S.: Stable and robust vectorization: How to make the right choices. In: Proceedings of Third IAPR Work. on Graphics Recognition. (1999) 3–16 Jaipur, India. 105

[9] Phillips, I., Chhabra, A.: Empirical performance evaluation of graphics recognition systems. IEEE Trans. on PAMI **21** (1999) 849–870. 105, 117

[10] Chhabra, A., Phillips, I.: The second international graphics recognition contest - raster to vector conversion: A report. In Tombre, K., Chhabra, A., eds.: Graphics Recognition: Algorithms and Systems. Springer, Berlin (1998) 390–410 Vol. 1389 of LNCS. 105, 117

[11] Chhabra, A., Philips, I.: Performance evaluation of line drawing recognition systems. In: Proceedings of 15th. Int. Conf. on Pattern Recognition. Volume 4. (2000) 864–869 Barcelona, Spain. 105, 117

[12] Groen, F., Sanderson, A., Schlag, F.: Symbol recognition in electrical diagrams using probabilistic graph matching. PRL **3** (1985) 343–350 . 106, 111, 126

[13] Jiang, X., Munger, A., Bunke, H.: Synthesis of representative graphical symbols by computing generalized median graph. In Chhabra, A., Dori, D., eds.: Graphics Recognition: Recent Advances. Springer-Verlag, Berlin (2000) 183–192 Vol. 1941 of LNCS. 106, 111, 115, 126

[14] Kuner, P., Ueberreiter, B.: Pattern recognition by graph matching. combinatorial versus continuous optimization. Int. Journal of Pattern Recognition and Artificial Intelligence **2** (1988) 527–542. 106, 111, 127

[15] Lee, S.: Recognizing hand-written electrical circuit symbols with attributed graph matching. In Baird, H., Bunke, H., Yamamoto, K., eds.: Structured Document Analysis. Springer Verlag, Berlin (1992) 340–358. 106, 111, 127

[16] Sato, T., Tojo, A.: Recognition and understanding of hand-drawn diagrams. In: Proceedings of 6th. Int. Conf. on Pattern Recognition. (1982) 674–677. 106, 127

[17] Messmer, B., Bunke, H.: Automatic learning and recognition of graphical symbols in engineering drawings. In Kasturi, R., Tombre, K., eds.: Graphics Recognition: Methods and Applications. Springer, Berlin (1996) 123–134 Vol. 1072 of LNCS. 106, 111, 127

[18] Muller, S., Rigoll, G.: Engineering drawing database retrieval using statistical pattern spotting techniques. In Chhabra, A., Dori, D., eds.: Graphics Recognition: Recent Advances. Springer-Verlag, Berlin (2000) 246–255 Vol. 1941 of LNCS. 106, 112, 127

[19] Lladós, J., Sánchez, G., Martí, E.: A string-based method to recognize symbols and structural textures in architectural plans. In Tombre, K., Chhabra, A., eds.: Graphics Recognition: Algorithms and Systems. Springer, Berlin (1998) 91–103 Vol. 1389 of LNCS. 106, 108, 127

[20] Valveny, E., Martí, E.: Hand-drawn symbol recognition in graphic documents using deformable template matching and a bayesian framework. In: Proceedings of 15th. Int. Conf. on Pattern Recognition. Volume 2. (2000) 239–242 Barcelona, Spain. 106, 108, 112, 127

[21] Chang, M., Chen, S.: Deformed trademark retrieval based on 2d pseudo-hidden markov model. PR **34** (2001) 953–967. 106, 108, 112, 126

[22] Cortelazzo, G., Mian, G., Vezzi, G., Zamperoni, P.: Trademark shapes descrition by string matching techniques. PR **27** (1994) 1005–1018. 106, 108, 126

[23] Segen, J.: From features to symbols: Learning relational models of shape. In Simon, J., ed.: From Pixels to Features. Elsevier Science Publishers B. V. (North-Holland) (1989) 237–248. 106, 111, 115, 127

[24] Bunke, H.: Attributed programmed graph grammars and their application to schematic diagram interpretation. IEEE Trans. on PAMI **4** (1982) 574–582. 106, 109, 111, 126

[25] Fahn, C., Wang, J., Lee, J.: A topology-based component extractor for understanding electronic circuit diagrams. IEEE Trans. on PAMI **2** (1989) 1140–1157. 111, 126

[26] Kiyko, V.: Recognition of objects in images of paper based line drawings. In: Proceedings of Third IAPR Int. Conf. on Document Analysis and Recognition, ICDAR'95. Volume 2., Montreal, Canada (1995) 970–973. 106, 111, 126

[27] Collin, S., Colnet, D.: Syntactic analysis of technical drawing dimensions. Int. Journal of Pattern Recognition and Artificial Intelligence **8** (1994) 1131–1148. 106, 107, 112, 126

[28] Dori, D.: A syntactic/geometric approach to recognition of dimensions in engineering machine drawings. Computer Vision, Graphics and Image Processing **47** (1989) 271–291. 107, 112, 126

[29] Joseph, S., Pridmore, T.: Knowledge-directed interpretation of mechanical engineering drawings. IEEE Trans. on PAMI **14** (1992) 928–940. 112, 126

[30] Min, W., Tang, Z., Tang, L.: Using web grammar to recognize dimensions in engineering drawings. PR **26** (1993) 1407–1916. 106, 107, 112, 127

[31] Fahmy, H., Blostein, D.: A survey of graph grammars: Theory and applications. In: Proceedings of 12th. Int. Conf. on Pattern Recognition (a). (1994) 294–298 Jerusalem, Israel. 106

[32] Sánchez, G., Lladós, J.: A graph grammar to recognize textured symbols. In: Proceedings of 6th Int. Conf. on Document Analysis and Recognition. (2001) Seattle, USA. 106, 112

[33] Kosmala, A., Lavirotte, S., Pottier, L., Rigoll, G.: On-Line Handwritten Formula Recognition using Hidden Markov Models and Context Dependent Graph Grammars. In: Proceedings of 5th Int. Conf. on Document Analysis and Recognition. (1999) 107 – 110 Bangalore, India. 106, 112, 126

[34] Lavirotte, S., Pottier, L.: Optical formula recognition. In: Proceedings of 4th Int. Conf. on Document Analysis and Recognition. (1997) 357 – 361 Ulm, Germany. 106, 109, 111, 127

[35] Bley, H.: Segmentation and preprocessing of electrical schematics using picture grafs. Computer Vision, Graphics and Image Processing **28** (1984) 271–288. 106, 112, 126

[36] Habacha, A.: Structural recognition of disturbed symbols using discrete relaxation. In: Proceedings of 1st. Int. Conf. on Document Analysis and Recognition. (1991) 170–178 Saint Malo, France. 106, 111, 126

[37] Vaxiviere, P., Tombe, K.: Celesstin: CAD conversion of mechanical drawings. Computer **25** (1992) 46–54. 106, 112, 128

[38] Myers, G., Mulgaonkar, P., Chen, C., DeCurtins, J., Chen, E.: Verification-based approach for automated text and feature extraction from raster-scanned maps. In Kasturi, R., Tombre, K., eds.: Graphics Recognition: Methods and Applications. Springer Verlag, Berlin (1996) 190–203. 106, 107, 112, 127

[39] Hartog, J., Kate, T., Gerbrands, J.: Knowledge-based segmentation for automatic map interpretation. In Kasturi, R., Tombre, K., eds.: Graphics Recognition: Methods and Applications. Springer, Berlin (1996) Vol. 1072 of LNCS. 106, 107, 112, 126

[40] Ah-Soon, C., Tombre, K.: Architectural symbol recognition using a network of constraints. PRL **22** (2001) 231–248. 106, 108, 112, 126

[41] Aoki, Y., Shio, A., Arai, H., Odaka, K.: A prototype system for interpreting hand-sketched floor plans. In: Proceedings of 13th. Int. Conf. on Pattern Recognition. (1996) 747–751 Vienna, Austria. 106, 108, 126

[42] Pasternak, B.: Processing imprecise and structural distroted line drawings by an adaptable drawing interpretation system. In Dengel, A., Spitz, L., eds.: Document Analysis Systems. World Scientific (1994) 349–365. 106, 112, 127

[43] Cheng, T., Khan, J., Liu, H., Yun, Y.: A symbol recognition system. In: Proceedings of Second IAPR Int. Conf. on Document Analysis and Recognition, ICDAR'93. (1993) 918–921 Tsukuba, Japan. 106, 110, 126

[44] Reiher, E., Li, Y., Donne, V., Lalonde, M., C.Hayne, Zhu, C.: A system for efficient and robust map symbol recognition. In: Proceedings of the 13th IAPR Int. Conf. on Pattern Recognition. Volume 3., Viena, Austria (1996) 783–787. 106, 107, 110, 127

[45] Anquetil, E., Coüasnon, B., Dambreville, F.: A symbol classifier able to reject wrong shapes for document recognition systems. In Chhabra, A., Dori, D., eds.: Graphics Recognition - Recent Advances. Springer, Berlin (2000) 209–218 Vol. 1941 of LNCS. 106, 107, 110, 126

[46] Miyao, H., Nakano, Y.: Note symbol extraction for printed piano scores using neural networks. IEICE Trans. Inf. and Syst. **E79-D** (1996) 548–553. 107, 110, 127

[47] Yadid-Pecht, O., Gerner, M., Dvir, L., Brutman, E., Shimony, U.: Recognition of handwritten musical notes by a modified neocognitron. Machine Vision and Applications **9** (1996) 65–72. 107, 110, 128
[48] Yang, D., Webster, J., Rendell, L., Garret, J., Shaw, D.: Management of graphical symbols in a cad environment: a neural network approach. In: Proceedings of Int. Conf. on Tools with AI. (1993) 272–279 Boston, Massachussets. 106, 110, 128
[49] Cesarini, F., Francesconi, E., Gori, M., Marinai, S., Sheng, J., Soda, G.: A neural-based architecture for spot-noisy logo recognition. In: Proceedings of Fourth IAPR Int. Conf. on Document Analysis and Recognition, ICDAR'97. Volume 1. (1997) 175–179 Ulm, Germany. 106, 108, 110, 126
[50] Francesconi, E., Frasconi, P., Gori, M., Mariani, S., Sheng, J., Soda, G., Sperduti, A.: Logo recognition by recursive neural networks. In Tombre, K., Chhabra, A., eds.: Graphics Recognition - Algorithms and Systems. Springer, Berlin (1998) Vol. 1389 of LNCS. 106, 108, 110, 126
[51] Kim, S., Suh, J., Kim, J.: Recognition of logic diagrams by identifying loops and rectilinear polylines. In: Proceedings of Second IAPR Int. Conf. on Document Analysis and Recognition, ICDAR'93. (1993) 349–352 Tsukuba, Japan. 106, 110, 126
[52] Parker, J., Pivovarov, J., Royko, D.: Vector templates for symbol recognition. In: Proceedings of 15th. Int. Conf. on Pattern Recognition. Volume 2. (2000) 602–605 Barcelona, Spain. 106, 110, 127
[53] Adam, S., Ogier, J., Cariou, C., Gardes, J., Mullot, R., Lecourtier, Y.: Combination of invariant pattern recognition primitives on technical documents. In Chhabra, A., Dori, D., eds.: Graphics Recognition - Recent Advances. Springer, Berlin (2000) 238–245 Vol. 1941 of LNCS. 106, 107, 110, 126
[54] Arias, J., Lai, C., Surya, S., Kasturi, R., Chhabra, A.: Interpretation of telephone system manhole drawings. PRL **16** (1995) 355–369. 107, 110, 126
[55] Boatto, L., Consorti, V., Del Buono, M., Di Zenzo, S., Eramo, V., Espossito, A., Melcarne, F., Meucci, M., Morelli, A., Mosciatti, M., Scarci, S., Tucci, M.: An interpretation system for land register maps. Computer **25** (1992) 25–33. 106, 107, 112, 126
[56] Samet, H., Soffer, A.: A legend-driven geographic symbol recognition system. In: Proceedings of 12th. Int. Conf. on Pattern Recognition (b). (1994) 350–355 Jerusalem, Israel. 127
[57] Samet, H., Soffer, A.: Marco: Map retrieval by content. IEEE Trans. on PAMI **18** (1996) 783–797. 107, 110, 127
[58] De Stefano, C., Tortorella, F., Vento, M.: An entropy based method for extracting robust binary templates. Machine Vision and Applications **8** (1995) 173–178. 106, 107, 110, 127
[59] Armand, J.: Musical score recognition: a hierarchical and recursive approach. In: Proceedings of Second IAPR Int. Conf. on Document Analysis and Recognition, ICDAR'93. (1993) 906–909 Tsukuba, Japan. 106, 107, 110, 126
[60] Doermann, D., Rivlin, E., Weiss, I.: Applying algebraic and differential invariants for logo recognition. Machine Vision and Applications **9** (1996) 73–86. 106, 108, 110, 116, 126
[61] Soffer, A., Samet, H.: Using negative shape features for logo similarity matching. In: Proceedings of 14th. Int. Conf. on Pattern Recognition (1). (1998) 571–573. 108, 110, 127
[62] Suda, P., Bridoux, C., Kammerer, Maderlechner, G.: Logo and word matching using a general approach to signal registration. In: Proceedings of Fourth IAPR

Int. Conf. on Document Analysis and Recognition, ICDAR'97. Volume 1. (1997) 61–65 Ulm, Germany. 106, 110, 127
[63] Lee, H., Lee, M.: Understanding mathematical expressions using procedure-oriented transformation. PR **27** (1994) 447–457. 106, 109, 110, 127
[64] Yu, Y., Samal, A., Seth, C.: A system for recognizing a large class of engineering drawings. IEEE Trans. on PAMI **19** (1997) 868–890. 106, 109, 110, 128
[65] Okazaki, A., Kondo, T., Mori, K., Tsunekawa, S., Kawamoto, E.: An automatic circuit diagram reader with loop-structure-based symbol recognition. IEEE Trans. on PAMI **10** (1988) 331–341. 106, 110, 127
[66] Jorge, J., Fonseca, M.: A simple approach to recognise geometric shapes interactively. In Chhabra, A., Dori, D., eds.: Graphics Recognition - Recent Advances. Springer, Berlin (2000) 266–274 Vol. 1941 of LNCS. 106, 109, 110, 126
[67] Yu, B.: Automatic understanding of symbol-connected diagrams. In: Proceedings of Third IAPR Int. Conf. on Document Analysis and Recognition, ICDAR'95. (1995) 803–806 Montreal, Canada. 106, 110
[68] Tombre, K., Dori, D.: Interpretation of engineering drawings. In Bunke, H., Wang, P., eds.: Handbook of character recognition and document image analysis. World Scientific Publishing Company (1997) 457–484. 106, 112, 127
[69] Antoine, D., Collin, S., Tombre, K.: Analysis of technical documents: The REDRAW system. In Baird, H., Bunke, H., Yamamoto, K., eds.: Structured document image analysis. Springer Verlag (1992) 385–402. 106, 107, 112, 126
[70] Madej, D.: An intelligent map-to-CAD conversion system. In: Proceedings of 1st. Int. Conf. on Document Analysis and Recognition. (1991) 602–610 Saint Malo, France. 106, 107, 112, 127
[71] Randriamahefa, R., Cocquerez, J., Fluhr, C., Pépin, F., Philipp, S.: Printed music recognition. In: Proceedings of Second IAPR Int. Conf. on Document Analysis and Recognition, ICDAR'93. (1993) 898–901 Tsukuba, Japan. 106, 107, 112, 127
[72] Ramel, J., Boissier, G., Emptoz, H.: A structural representation adapted to handwritten symbol recognition. In Chhabra, A., Dori, D., eds.: Graphics Recognition: Recent Advances. Springer-Verlag, Berlin (2000) 228–237 Vol. 1941 of LNCS. 106, 109, 112, 127
[73] Kasturi, R., Bow, S., El-Masri, W., Shah, J., Gattiker, J., Mokate, U.: A system for interpretation of line drawings. IEEE Trans. on PAMI **12** (1990) 978–992. 106, 109, 112, 126
[74] Ventura, A., Schettini, R.: Graphic symbol recognition using a signature technique. In: Proceedings of 12th. Int. Conf. on Pattern Recognition (b). (1994) 533–535 Jerusalem, Israel. 106, 128
[75] Wilfong, G., Sinden, F., Ruedisueli, L.: On-line recognition of handwritten symbols. IEEE Trans. on PAMI **18** (1996) 935–940. 106, 109, 128
[76] Ablameyko, S.: An Introduction to Interpretation of Graphic Images. SPIE Optical Engineering Press (1997) 106
[77] Dosch, P., Masini, G., Tombre, K.: Improving arc detection in graphics recognition. In: Proceedings of 15th. Int. Conf. on Pattern Recognition. Volume 2. (2000) 243–246 Barcelona, Spain. 106
[78] Ablameyko, S., Bereishik, V., Frantskevich, O., Homenko, M., Paramonova, N.: Knowledge-based recognition of crosshatched areas in engineering drawings. In Amin, A., Dori, D., Pudil, P., Freeman, H., eds.: Advances in Pattern Recognition. Vol. 1451 of LNCS (1998) 460–467. 106
[79] Lladós, J., Martí, E., López-Krahe, J.: A Hough-based method for hatched pattern detection in maps and diagrams. In: Proceedings of 5th Int. Conf. on Document Analysis and Recognition. (1999) 479–482 Bangalore, India. 107

[80] Fahmy, H., Blonstein, D.: A graph grammar programming style for recognition of music notation. Machine Vision and Applications **6** (1993) 83–99. 107, 111, 126

[81] Landay, J., Myers, B.: Sketching interfaces: Toward more human interface design. IEEE Computer **34** (2001) 56–64. 109, 127

[82] Wenyin, L., Jin, X., Qian, W., Sun, Z.: Inputing composite graphic objects by sketching a few constituent simple shapes. In: Proceedings of Fourth IAPR Work. on Graphics Recognition. (2001) 73–84 Kingston, Canada. 109

[83] Blostein, D., et al.: User interfaces for on-line diagram recognition. In: Proceedings of Fourth IAPR Work. on Graphics Recognition. (2001) 95–106 Kingston, Canada. 109

[84] Paek, S., Smith, J.: Detecting image purpose in world-wide documents. In Lopresti, D., Zhou, J., eds.: Document Recognition V. SPIE, Bellingham, Whashington, USA (1998) 151–158 Vol. 3305 of Proceedings of SPIE. 109

[85] Duda, R., Hart, P., Stork, D.: Pattern Classification and Scene Analysis. John Wiley and Sons, New York (2000) 109

[86] Fu, K.: Syntactic Pattern Recognition and Applications. Prentice-Hall, Englewood Cliffs, N. J. (1982) 109

[87] Jain, A., Duin, R., Mao, J.: Statistical pattern recognition: a review. IEEE Trans. on PAMI **22** (2000) 4–37. 109

[88] Pavlidis, T.: Structural Pattern Recognition. Springer-Verlag, New York (1977) 109

[89] Trier, O., Jain, A., Taxt, T.: Feature extraction methods for character recognition - a survey. PR **29** (1996) 641–662. 110

[90] Furuta, M., Kase, N., Emori, S.: Segmentation and recognition of symbols for handwritten piping and instrument diagram. In: Proceedings of the 7th IAPR Int. Conf. on Pattern Recognition. (1984) 626–629. 110

[91] Lin, X., Shimotsuji, S., Minoh, M., Sakai, T.: Efficient diagram understanding with characteristic pattern detection. Computer Vision, Graphics and Image Processing **30** (1985) 84–106. 110

[92] Lladós, J.: Combining Graph Matching and Hough Transform for Hand-Drawn Graphical Document Analysis. Application to Architectural Drawings. PhD thesis, Universitat Autònoma de Barcelona and Université de Paris 8 (1997) 111

[93] Burr, D.: Elastic matching of line drawings. IEEE Trans. on PAMI **3** (1981) 708–713. 112

[94] Doermann, D.: The indexing and retrieval of document images: A survey. Technical report, University of Maryland (1998) Technical Report CS-TR-3876. 113

[95] Casacuberta, F., de Antonio, M.: A greedy algorithm for computing approximate median strings. In: VII Spanish Simposium on Pattern Recognition and Image Analysis. (1997) 193–198 Barcelona. 115

[96] Wong, A., You, M.: Entropy and distance of random graphs with application to structural pattern recognition. IEEE Trans. on PAMI **7** (1985) 599–609. 115

[97] Cordella, L., Foggia, P., Genna, R., Vento, M.: Prototyping structural descriptions: an inductive learning approach. In Amin, A., Dori, D., Pudil, P., Freeman, H., eds.: Advances in Pattern Recognition. Springer Verlag, Berlin (1998) 339–348. 116

[98] Valveny, E., Martí, E.: Learning structural descriptions of graphic symbols using deformable template matching. In: Proceedings of 6th Int. Conf. on Document Analysis and Recognition. (2001) Seattle, USA. 116

[99] Messmer, B.: Efficient Graph Matching Algorithms for Preprocessed Model Graphs. PhD thesis, University of Bern (1995) 116

[100] Sossa, H., Horaud, R.: Model indexing: The graph-hashing approach. In: Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition. (1992) 811–814 Champaign, Illinois. 116

[101] Liu, W., Dori, D.: A protocol for performance evaluation of line detection algorithms. Machine Vision and Applications **9** (1997) 240–250. 117

[102] Liu, W., Dori, D.: A proposed scheme for performance evaluation of graphics/text separation algorithms. In Tombre, K., Chhabra, A., eds.: Graphics Recognition: Algorithms and Systems. Springer, Berlin (1998) 359–371 Vol. 1389 of LNCS. 117

[103] Aksoy, S., Ye, M., Schauf, M., Song, M., Wang, Y., Haralick, R., Parker, J., Pivovarov, J., Royko, D., Sun, C., Farneboock, G.: Algorithm performance contest. In: Proceedings of 15th. Int. Conf. on Pattern Recognition. Volume 4. (2000) 870–876 Barcelona, Spain. 117

[104] Yamada, H., Yamamoto, K., Hosokawa, K.: Directional mathematical morphology and reformalized Hough transformation for the analysis of topographic maps. IEEE Trans. on PAMI **15** (1993) 380–387. 128

A Bibliography Summary

Table 3. Bibliography summary

Reference	Application	Segmentation	Primitives	Recognition method	Notes
Adam:00 [53]	Utility maps	Connected components	Fourier-Melin transform, Zernike moments, circular primitives	Learning vector quantization, 1NN	
Ah-Soon:01 [40]	Architecture	Integrated with recognition	Vectors	Network of constraints	
Anquetil:00 [45]	Musical scores	Connected components	Pixels	Neural network	
Antoine:92 [69]	Cadastral maps	Integrated with recognition	Vectors	Specific for each type of symbol	
Aoki:96 [41]	Architecture	Low level: segmentation of lines and geometric primitives	Low level geometric entities (squares, circles, lines, etc.)	Pattern matching (low-level primitives) and rules (symbols)	
Arias:95 [54]	Telephone maps	Integrated with recognition	Vectors	Pattern matching	
Armand:93 [59]	Musical scores	Specific to isolate notes and staff lines	Geometric feature vector	Nearest Neighbour	
Bley:84 [35]	Logic diagrams	Based on run-length encoding	Graph	Production system	
Boatto:92 [55]	Cadastral maps	Based on run-length encoding	Vectors and geometric features	Similarity in terms of geometric and topologic features	
Bunke:82 [24]	Logic diagrams and flowcharts	Integrated with recognition	Graph	Graph grammar	Distortion model that allows hand drawn diagrams
Cesarini:97 [49]	Logos	Connected components	Image normalized to a fixed size	Autoassociative neural networks	
Chang:01 [21]	Logos	Assumes presegmented symbols	Log-polar space	2D HMM	
Cheng:93 [43]	Logic diagrams	User driven	Invariant moments	Hierarchic neural network	
Collin:94 [27]	Engineering drawings	Assumes presegmented symbols	Lines, arrowheads, etc.	Plex-grammars	
Cortelazzo: 94 [22]	Logos	Assumes presegmented symbols	Strings	String matching	
Doermann: 96 [60]	Logos	Text-graphics separation	Contours	Comparison of algebraic invariants from boundary	
Dori:89 [28]	Engineering drawings	Assumes presegmented symbols	Graph	Web grammars (dimensions) and lookup table (low level primitives)	
Fahmy:93 [80]	Musical scores	Assumes presegmented symbols	Graph	Graph grammar	
Fahn:89 [25]	Logic diagrams	Integrated with recognition	Vectors	Non contextual grammar	
Francesconi: 98 [50]	Logos	Assumes presegmented symbols	Tree of boundary segments with geometric attributes	Recursive neural network	
Groen:85 [12]	Logic diagrams	Based on loops	Graph	Probabilistic graph matching	
Habacha:91 [36]	Logic diagrams	Integrated with recognition	Graph	Discrete relaxation	
Hartog:96 [39]	Utility maps	Integrated with recognition	Pixels	Rule-based system	
Jiang:00 [13]	Logic diagrams	Assumes presegmented symbols	Graph	Graph Matching and genetic algorithm	Symbol learning by mean graph computation
Jorge:00 [66]	On-line diagrams	Based on the sketches and time-outs	Similarity between geometric features	Decision trees with fuzzy logics	
Joseph:92 [29]	Engineering drawings	Integrated with recognition	Vectors	Recognition driven by a grammar	The grammatical rules are the control mechanism that guides the recognition strategy
Kasturi:90 [73]	Line drawings	Based on minimum redundancy loops	Graph	Specific for each type of symbol	
Kim:93 [51]	Logic diagrams	Based on loops	Fourier descriptors and moments	Nearest Neighbour	
Kiyko:95 [26]	Logic diagrams	Integrated with recognition	Skeleton	Grammar	Also applied to hand drawn sketches
Kosmala:99 [33]	Mathematical formulas	Connected components	Pixels	HMM and graph grammar	

Table 3. (continued)

Reference	Application	Segmentation	Primitives	Recognition method	Notes
Kuner:88 [14]	Logic diagrams	The input image is divided in windows	Skeleton graph	graph matching and rule-based system to solve ambiguities	
Landay:01 [81]	User Interfaces	Assumes presegmented symbols	Sequence of coordinates	Statistical PR	On-line recognition
Lavirotte:97 [34]	Mathematical formulas	Connected components	Pixels	graph grammar	
Lee:92 [15]	Logic diagrams	Assumes presegmented symbols	Graph	graph matching	
Lee:94 [63]	Mathematical formulas	Connected components	Feature vectors	Nearest Neighbour and dynamic programming to modelize distortion	
Llados:98 [19]	Architecture	Integrated with recognition	Graph	Error-tolerant graph matching	Applied to hand drawn diagrams
Madej:91 [70]	Cadastral maps	Based on symbol-dependent heuristics	Hierarchical graph	Heuristic	
Messmer:96 [17]	Engineering drawings	Integrated with recognition	Vectors	Graph Matching	Symbol models are previously compiled in a network to reduce the computational cost of recognition
Min:93 [30]	Engineering drawings	Assumes presegmented symbols	Arrowheads, lines, etc.	Web grammar	
Miyao:96 [46]	Musical scores	Combines projections, runs and meshes	Feature vectors	Neural network	
Muller:00 [18]	Engineering drawings	Integrated with recognition	Cosine transform	2D Hidden Markov Model	Database retrieval by sketching
Myers:96 [38]	Geographic maps	Hypotheses generation concerning to image locations likely to contain symbols	Pixels	hypothesis-and-test	
Okazaki:88 [65]	Logic diagrams	Based on lines and connections	Primitives found by pattern matching	Decision tree	
Parker:00 [52]	Logic diagrams	Assumes presegmented symbols	Vectors	Template matching	Also applied to hand drawn textual symbols
Pasternak:94 [42]	General	Integrated with recognition	Arcs and lines	Driven by a language of descriptions	
Ramel:00 [72]	Formula	Integrated with recognition	Graph	Heuristic	
Randriamahefa: 93 [71]	Musical scores	Based on horizontal lines and connected components	Attributed graph	Rule-based system	
Reiher:96 [44]	Geographic maps	Integrated with recognition	Pixels	Hausdorff distance and neural network	
Samet:94 [56]	Geographic maps	Based on color information	Pixels	k-nearest neighbours	The recognition is driven by the information found in the legend
Samet:96 [57]	Geographic maps	Connected components	Globals (Moments, circularity, eccentricity) and locals (crossings, gaps)	Voting by weighted bounded several-nearest neighbor classifier	
Sato:82 [16]	Electronic diagrams	"Image lines are followed using a window to separate long lines and "complex regions"	Lines and regions	Relational matching	
Segen:89 [23]	Shapes	Assumes presegmented symbols	Boundary	Bondary saliency points comparison	
Soffer:98 [61]	Logos	Assumes presegmented symbols	Connected components	Comparison of connected component attributes	Indexing in document databases
Stefano:95 [58]	Geographic maps	Integrated with recognition	Pixels	Template matching	
Suda:97 [62]	Logos	Assumes presegmented symbols	Image sampling using a fixed grid	Distance between feature vectors (zoning)	
Tombre:97 [68]	Engineering drawings	Integrated with recognition	Vectors	Specific for each type of symbol	
Valveny:00 [20]	Architecture	Assumes presegmented symbols	Pixels and vectors	Deformable models	Also applied to hand drawn symbols

Table 3. (continued)

Reference	Application	Segmentation	Primitives	Recognition method	Notes
Vaxiviere:92 [37]	Engineering drawings	Integrated with recognition	Vectors	Rule-based system	
Ventura:94 [74]	CAD	Integrated with recognition	Vectors and background areas	Signature indexing	
Wilfong:96 [75]	Characters	Assumes presegmented symbols	Sequence of coordinates	Curvature distance	On-line recognition
Yadid-Pecht:96 [47]	Musical scores	Specific to isolate notes and staff lines	Pixels	"Neural networks; Neocognitron"	
Yamada:93 [104]	Topographic maps	Integrated with recognition	Pixels	Hough Transform approach combined with directional morphologic operators	The recognition is driven by the information found in the legend
Yang:93 [48]	Architecture	The drawing is analyzed dynamically	Pixels	Neural network	
Yu:97 [64]	Line drawings	Using rules on the lines properties to separate them from connections	Loops	Hierarchical matching	