

Integrating vocabulary clustering with spatial relations for symbol recognition

K. C. Santosh · Bart Lamiroy · Laurent Wendling

Received: 13 June 2012 / Revised: 20 April 2013 / Accepted: 22 May 2013
© Springer-Verlag Berlin Heidelberg 2013

Abstract This paper develops a structural symbol recognition method with integrated statistical features. It applies spatial organisation descriptors to the identified shape features within a fixed visual vocabulary that compose a symbol. It builds an attributed relational graph expressing the spatial relations between those visual vocabulary elements. In order to adapt the chosen vocabulary features to multiple and possible specialised contexts, we study the pertinence of unsupervised clustering to capture significant shape variations within a vocabulary class and thus refine the discriminative power of the method. This unsupervised clustering relies on cross-validation between several different cluster indices. The resulting approach is capable of determining part of the pertinent vocabulary and significantly increases recognition results with respect to the state-of-the-art. It is experimentally validated on complex electrical wiring diagram symbols.

Keywords Spatial relations · Visual vocabulary · Shape descriptor · Unsupervised clustering · Symbol recognition

1 Introduction

Graphics recognition has an extremely rich state-of-the-art literature in symbol recognition and localisation. However, most methods are targeted towards isolated line symbols, not for composed symbols connected to a complex environment [12,27]. Considering the problem of symbol localisation in real documents, composed of individual parts and constrained by spatial relations for instance, one needs to be able to extract visual parts, characterise their shape description and formalise the possible links that exist between them. This integration of spatial relations and shape description of the extracted visual parts is going to be the core of this paper.

Among the existing state-of-the-art, structural approaches provide powerful representations, conveying how parts are connected to each other. However, relations [38,44] do not exploit shape information as shape descriptors do. On the other hand, especially when symbols are not in simple linear and isolated form, but are composed of many elements, sometimes exhibiting subtle differences, global signal-based shape descriptors cannot provide optimal retrieval performance.

In this paper, we aim to combine both structural and statistical approaches and try to avoid the shortcomings of each of them. To do so, we decompose symbols by expressing their various parts in a fixed visual vocabulary, using spatial relations, graphs and signal-based descriptors to describe the whole shape. This paper is the extension of previous work [41,44,51] where we have established the significance of spatial relations for symbol recognition.

1.1 Related work

Studies related to the inclusion of spatial relations for symbol recognition, symbol representation and matching techniques as well as the performance analysis of several different struc-

K. C. Santosh (✉) · B. Lamiroy
LORIA (UMR 7503), Université de Lorraine,
Campus Scientifique, BP 239, 54506 Vandoeuvre-lès-Nancy
Cedex, France
e-mail: Santosh.KC@loria.fr

B. Lamiroy
e-mail: Bart.Lamiroy@loria.fr

L. Wendling
SIP, LIPADE, Université Paris Descartes (Paris V),
45, rue des Saints-Pères, 75270 Paris Cedex 06, France
e-mail: Laurent.Wendling@parisdescartes.fr

tural approaches can be found in Cordella and Vento [11, 12], Lladós et al. [26, 27]. It seems to us that some of their limitations may be addressed by looking at their possible integration with statistical approaches.

In Cordella and Vento [11], shape analysis for symbol recognition has been comprehensively addressed. Their context mainly consists of isolated binary shapes. In parallel, statistical approaches like global signal-based descriptors [4, 23, 47, 59–61] are usually quite fault tolerant to image distortions, since they tend to filter out small detail changes. This is unfortunately inconvenient in our context where symbols may either be very similar in shape—and only differ by slight details—or either be completely different from a visual point of view. Symbols may also be composed of other known and significant symbols and need not necessarily be connected. Moreover, the previously mentioned methods difficultly accommodate with connected or composite symbols and their major drawbacks are due to deformation, composition with other symbols (which, in Yuen et al. [59] leads to unstable centroid detection, and thus errors in the ring projection) and occlusion over the boundary (leading to unstable tangents in shape context (SC) [4], for instance). In some cases, researchers have been integrating descriptors [3, 40, 48] or have been combining several classifiers [49] to increase their performance, partially based on the idea presented in Tombre et al. [52] that off-the-shelf methods are primarily designed for applications where line symbols are isolated. In these statistical approaches, signatures are simple with low computational cost. However, discrimination power and robustness strongly depend on the selection of an optimal set of features for each specific application.

Besides global shape-based symbol description, another idea is to decompose the symbols into either vector-based primitives like points, lines, arcs, etc., or into meaningful parts like *circles*, *triangles*, *rectangles*, etc. These methods fall under structural approaches. They are then represented as attributed relational graphs (ARG) [8, 10], region adjacency graphs (RAG) [26], constraint networks [1] as well as deformable templates [54]. Their common drawback comes from error-prone raster-to-vector conversion. Those errors can increase confusions among different symbols. Furthermore, variability of the size of graphs leads to computational complexity in matching. However, structural approaches provide a powerful representation, conveying how parts are connected to each other, while also preserving generality and extensibility. Several other approaches are more focused on computing symbol signatures by taking some regions of interest in the document image [15, 37, 56]. These methods aim at providing faster matching in comparison with graph matching. On the other hand, they are dependent on the region of interest detector.

The overall conclusion is that one needs an appropriate image description so that the advantages of statistical fea-

tures can be integrated with the expressiveness of structural approaches, thus providing generality and extensibility properties. This was already mentioned in Tombre [50]:

... the very structural and spatial nature of the information we work with makes structural methods quite natural in the community. Their efficient integration into methods which also take full advantage of statistical learning and classification is certainly the right path to take.

An interesting example that uses shape descriptions and relations to form a RAG is found in Bodic et al. [7]. The vector-based RAG is based on segmented regions which are labelled as vertices and geometric properties of adjacency relations are used to label edges. However, the approach is limited once segmented regions change with image transformations.

In the framework of stroke-based hand-drawn symbol recognition, two studies are interesting to consider [22, 25]. The first one is related to template-based matching. The other one uses an ARG where the vertices represent geometric primitives like lines and arcs, and the edges represent the geometric relationships between them. Matching is primarily based on graph matching or graph isomorphism [29] and is conceptually similar to [57]. These approaches perform well as long as the vertices are well separated and segmented (which is the case, since they are taken from online strokes and thus vectorisation difficulties are avoided). Recently, Coustaty et al. [13] introduced an interesting approach where a Galois lattice is used to classify structural signatures extracted by using the Hough transform. These structural signatures are based on a topological graph using five topological relations computed between the segments as well as their lengths (based on connected and disconnected topological configurations). The paper reports that Galois lattice-based classification is very robust to noise. This, however, may not provide consistent performance when symbols are found with other graphical elements or with possible texts within the image. Furthermore, since it is very dependent on the Hough transform-based segment extraction, it cannot easily incorporate more statistical shape descriptions and therefore is not suitable for our study.

Our approach is to integrate shape descriptors with spatial relations between the visual primitives that compose the symbol. In what follows, an outline of the proposed method will be explained.

1.2 Outline of the proposed method

Global signal-based descriptors can only be applied to isolated patterns. Bag-of-words approaches are widely used to overcome this problem, but require extensive training sets on

the one hand, and do not take into account the global structure or arrangements between the extracted visual words. Furthermore, they usually lose the human-intuitive visual semantics of the symbol. In cases where not many training samples are available, or too costly to obtain; where the symbols' visual data itself are very redundant and overlapping, and where it is important that the symbol description matches a human sense of semantics, bag-of-words approaches are ill-suited. In previous work, we addressed some of these issues [44] and studied the possibility of having human semantics centred approaches [41].

We assume that there is no extensive training set available (in our experimental setup, we use a catalogue of known symbols for aircraft electric wiring diagrams [53], for which only one single instance for each symbol class is available) but that there is enough human expertise available to provide knowledge of what is considered discriminant between the various symbols. Therefore, the main idea behind the approach developed in this paper is to use a set of well defined, robust, high level visual part extractors, segmenting shapes into these discriminant elementary parts, we will refer to as *visual vocabulary*. The driving motivation behind this is that a library of well mastered, robust and generic extraction tools can replace statistical bag-of-words learning techniques when insufficient learning data are available, or as we shall show in Sect. 2.2, if the generic tools are not discriminant enough, it can be combined with unsupervised clustering techniques, to improve their discriminative power.

Once the symbols are segmented into their vocabulary parts, we can use the spatial relations between them [44] to express the global spatio-structural information of the shape. However, these relations do not express shape information in the same way shape descriptors do. The spatial relation descriptors express global pixel distributions between identified areas. The areas themselves are the result of the previously mentioned vocabulary extractors.

Figure 1 depicts how our system operates. First visual primitives are extracted from a symbol. On the one hand,

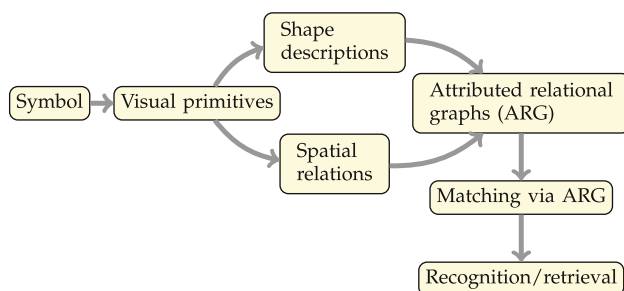


Fig. 1 An architecture of the proposed method that addresses structural method, integrated with statistical features. It uses ARG-based symbol description using spatial organisation and shape features of the visual vocabulary that compose a symbol

they give structural information of the elementary shapes they represent. On the other hand, computing spatial relations expresses how these different shapes are positioned to one another within the symbol. The shapes are taken as nodes and the relations as arcs in an attributed relational graph representing the symbol. The recognition process relies on graph matching (which in this case is fairly trivial, as explained in Sect. 2, since all graph nodes are uniquely labelled and all instances of one specific vocabulary type are merged into one single node).

Unfortunately, our first experiments showed limitations when the vocabulary extraction operators were too broadly defined. The core of this paper is therefore to address this problem by introducing unsupervised clustering on the broad visual vocabulary classes, in order to refine them into more visually discriminant subclasses. This is made possible by the fact that, once segmented from their initial images, they form a sufficient number of training samples.

The remaining of the paper is organised as follows. We start by explaining our symbol description in Sect. 2, and more specifically, how spatial relation distributions are used as arc attributes between two vertices in our ARG framework. We then discuss unsupervised vocabulary clustering via shape analysis, allowing us to subdivide nodes into discriminant sub-shapes. In Sect. 3, our symbol recognition process is explained. Full experiments are reported in Sect. 4 and provides details of our unsupervised clustering and its validation by using several different cluster validation indices. The paper is concluded in Sect. 5.

2 Symbol description

This section describes the general symbol description approach we use: first, we describe the visual vocabulary, how it is obtained and the resulting ARG. Second, we extend this approach by introducing a more refined analysis of the visual data segmented from the vocabulary, in order to automatically detect visually coherent subclasses, using unsupervised classification.

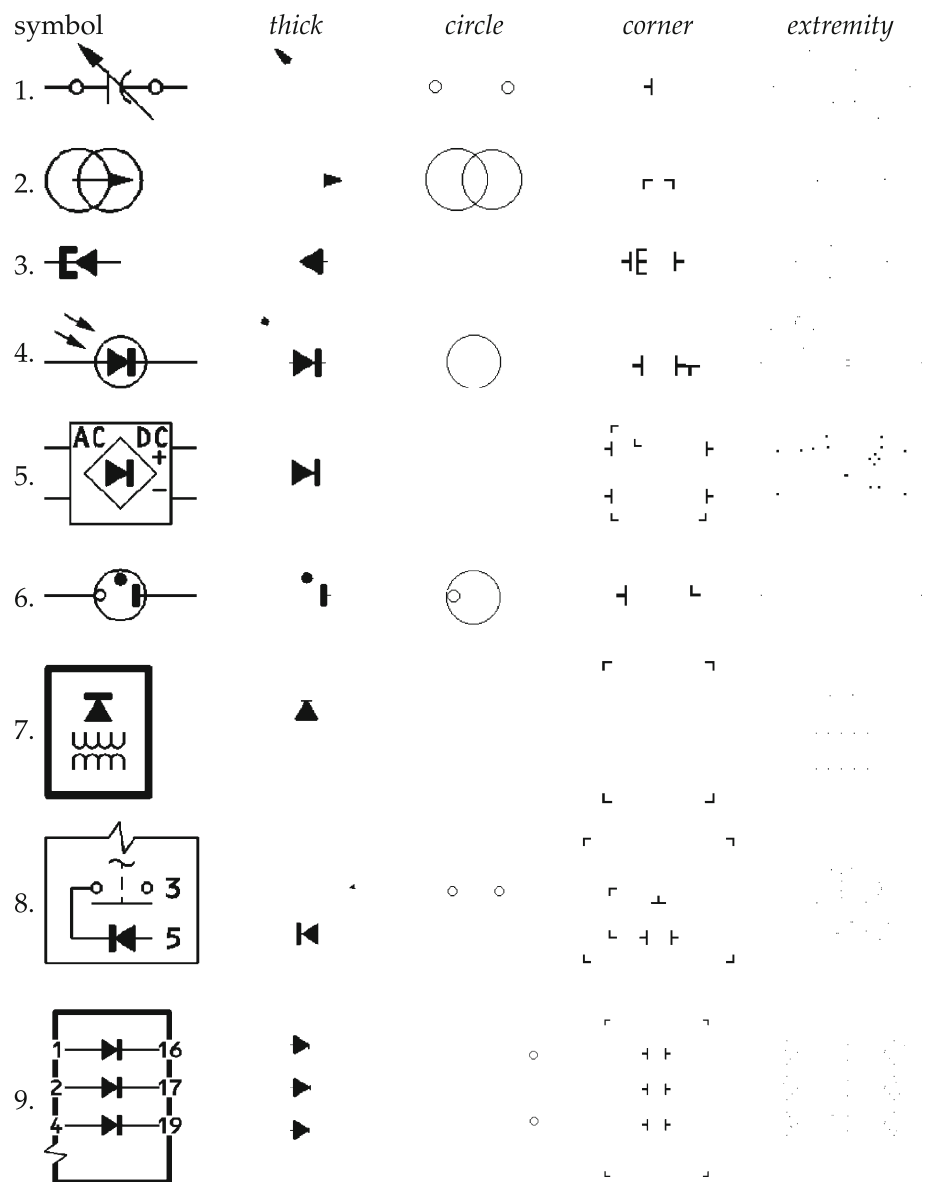
2.1 Graph via visual vocabulary

As explained before, we are going to construct an ARG based on extracted visual components, linked together with their relative spatial distribution relations.

2.1.1 Visual vocabulary

While, in the general case, the extracted vocabulary can be of any kind and from any type of features, related to what is visually pertinent in the application context under consideration, our current vocabulary is related to electrical symbols. It

Fig. 2 Visual vocabulary from a few corresponding symbols



can be easily extended or modified by using different vocabularies and other visual cues to adapt to other domains.

We define a set of well-controlled visual primitives as a *vocabulary*. They are extracted with the help of classical image analysis operators [16, 24, 33]. Our vocabulary set consists of *circles*, *corners*, loose end *extremities* and *thick* (filled) components. Figure 2 shows a few examples.

1. Circle primitive

We use the algorithm as described in Lamiroy and Guebhas [24] which is based on random sample consensus minimisation.

2. Corner primitive

We only consider straight angle corners. They are extracted using a simple template matching process. If the

ratio of black and white pixels is greater than or equal to the template threshold, then the presence of corner is assessed.

3. Extremity primitive

We detect loose ends from the image skeleton pixels.

4. Thick primitive

We employ straightforward thin/thick separation by counting all *thick* connected components within the image. It takes place in a two-step process:

- it uses standard skeletonisation using chamfer distance and computes the histogram of line thicknesses; and
- an optimal cut value is computed from the histogram to distinguish between thick zones and thin zones.

On the whole, our current implementation is based on [16, 33] that uses a straightforward histogram high-pass filter, following the line thickness in the document image.

In what follows, we shall refer to the set of vocabulary types as,

$$\sum_{\mathbb{T}} = \{\mathbb{T}_{\text{thick}}, \mathbb{T}_{\text{circle}}, \mathbb{T}_{\text{corner}}, \mathbb{T}_{\text{extremity}}\}. \quad (1)$$

2.1.2 Graph-based representation

Rather than using the detected elements as a basis for expressing and computing spatial relations, we group them together according to their types, as shown in Fig. 2. In a sense, a symbol is decomposed into (in our case) four layers, each of which represents either all thick components, all circles, all corners or all extremities.

The symbol can then be represented by a complete ARG as a 4-tuple $G = (V, E, F_A, F_E)$ where

- V is the set of vertices,
- $E \subseteq V \times V$ is the set of graph edges,
- $F_A : V \rightarrow \sum_{\mathbb{T}}$ is a function assigning attributes to the vertices. The assigned attribute is one of the extracted vocabulary classes.
- $F_E : E \rightarrow \mathcal{R}_E$ is a function assigning labels to the edges where \mathcal{R} represents spatial relations of the edge E as developed in Santhosh et al. [44].

Effects of spatial relations on recognition performance have already been examined comprehensively for scene understanding [2, 5], document analysis and recognition tasks [32, 38, 44, 57]. Based on [19], spatial relations can be

1. topological [18] such as *disconnected*, *externally connected*, *overlap* and *contain or inside*;
2. directional [6, 28, 55] such as *north* and *south*; and
3. metric such as *near* or *far*.

Previously mentioned approaches address only either topological or directional relations. Managing both comes at high computational costs. Even then, no existing model fully integrates topology. Our approach [42, 44] unifies both topological and directional information into one descriptor without any additional running time cost. Given two nodes,

1. it first uses their topological configuration to find a unique reference point;
2. from that reference point, a radial line is rotated over a cycle with a certain angular step, in order to cover a surrounding space; and




3. a directional relational histogram is computed at each angular step, accounting for the percentage of pixels of the studied visual primitives lying in that direction.

More formally, the proposed graph can be expressed as $G = \{$

$$\begin{aligned} V &= \{\mathbb{T}_1, \mathbb{T}_2, \mathbb{T}_3, \mathbb{T}_4\}, \\ E &= \{(\mathbb{T}_1, \mathbb{T}_2), (\mathbb{T}_1, \mathbb{T}_3), \dots, (\mathbb{T}_3, \mathbb{T}_4)\}, \\ F_A &= \{(\mathbb{T}_1, \mathbb{T}_{\text{thick}}), (\mathbb{T}_2, \mathbb{T}_{\text{circle}}), (\mathbb{T}_3, \mathbb{T}_{\text{corner}}), (\mathbb{T}_4, \mathbb{T}_{\text{extremity}})\}, \\ F_E &= \{((\mathbb{T}_1, \mathbb{T}_2), \mathcal{R}_{\mathbb{T}_1, \mathbb{T}_2}), ((\mathbb{T}_1, \mathbb{T}_3), \mathcal{R}_{\mathbb{T}_1, \mathbb{T}_3}), ((\mathbb{T}_2, \mathbb{T}_3), \mathcal{R}_{\mathbb{T}_2, \mathbb{T}_3})\}. \end{aligned} \quad (2)$$

Since this forms a complete graph, it is obvious that there exist $r = \frac{t(t-1)}{2}$ edges for t attribute types. The major intention of having fixed and completely labelled attributes is to avoid the NP-hardness of the matching problem [44] and also to keep coherence as vocabulary elements are semantically different.

2.1.3 Limitations and graph extension

Due to shape and size variation of the *thick* patterns, the discriminative power of this description (and therefore, the resulting retrieval performance) is sub-optimal [44]. Taking a closer look at the extracted *thick* patterns from different symbols (Fig. 3), it is obvious that the shape and size of the *thick* pattern is related to category of the symbol from which it is assumed to be extracted. For instance, a *thick* pattern  coming from a junction is different from a triangle-shaped one, such as a significant part of a diode symbol or from an arrow: , . Therefore, and in order to better distinguish these cases in the current ARG framework [(cf. Eq. (2))], we integrate shape descriptors and apply the shape features to label vertices.

$$F_A = \{(\mathbb{T}_1, \mathbb{T}_{\text{thick}}), \mathcal{S}_{\mathbb{T}_1}\}, (\mathbb{T}_2, \mathbb{T}_{\text{circle}}), (\mathbb{T}_3, \mathbb{T}_{\text{corner}}), (\mathbb{T}_4, \mathbb{T}_{\text{extremity}})\}. \quad (3)$$

Representing a single complete vertex via a global signal-based descriptor does not sufficiently exploit all available information since it is a collection of similar vocabulary type instances. We therefore introduce the following approach: since the number of *thick* patterns can be different from one symbol to another, the vertex labelled with the *thick* vocabulary type will be split into more specialised *thick* sub-vertices. This is done separately for all individual *thick* patterns.

$$\mathcal{S}_{\mathbb{T}_1} = \{s_{\mathbb{T}_1, \kappa}, \dots, s_{\mathbb{T}_1, \mathcal{K}}\}, \quad (4)$$

where \mathcal{K} is the number of *thick* patterns in a symbol. For any symbol, there thus obtain $\{G_{\kappa}\}_{\kappa=1, \dots, \mathcal{K}}$ ARGs. Figure 4 shows a description of the proposed approach.

Fig. 3 Symbols and their corresponding *thick* patterns. For better understanding, extracted *thick* patterns are enlarged

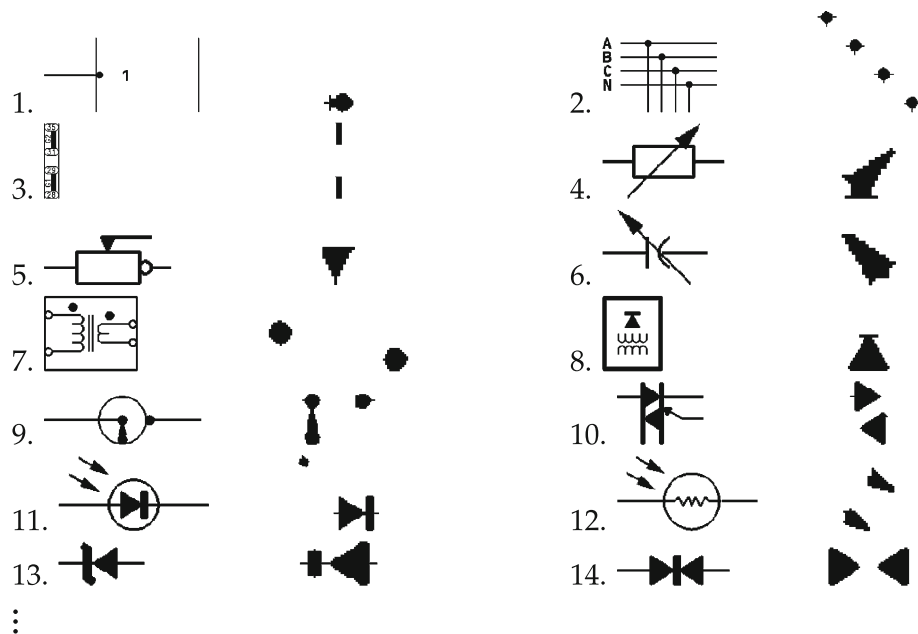
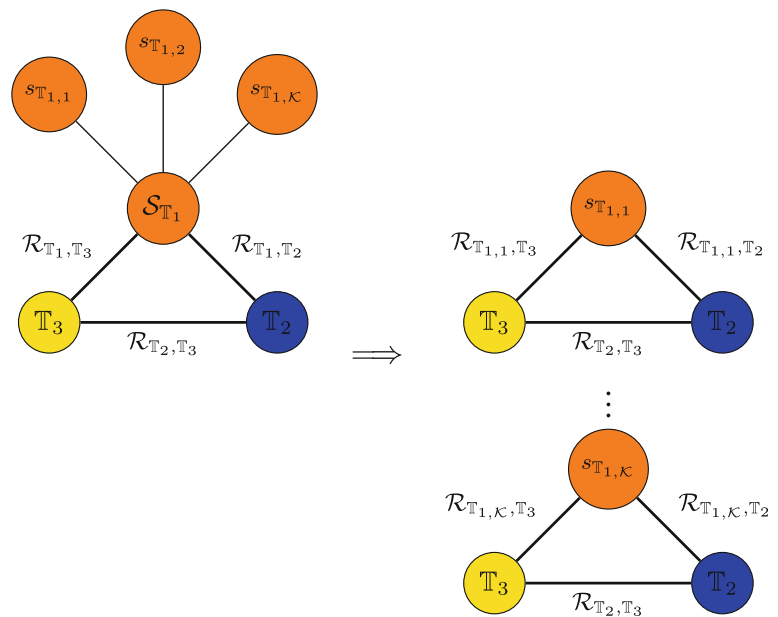


Fig. 4 A complete ARG description for a symbol with vocabulary type set *thick*, *corner* and *extremity*. The original graph is split further into a number of *graphs* corresponding to the number of extracted *thick* subclasses



According to our description (cf. Fig. 4), matching time will basically be increased in accordance with the number of *thick* patterns that compose the symbol. To reduce such a high time processing, we propose to use *thick* pattern clustering. The main idea developed in the next section is to refine the notion of *thick* patterns, by collectively considering all components extracted from the symbol set, to compute individual shape descriptors on each of the components and to apply unsupervised clustering to generate a set of visually similar subclasses within the *thick* pattern class as illustrated in Fig. 4.

2.2 Shape-based *thick* pattern clustering

As said before, the main idea is to perform *thick* patterns clustering using shape descriptors so that patterns with different global appearances will fall under different clusters or groups. The collected *thick* patterns in a cluster are assumed to be extracted from similar types of symbols. Since we have no a priori knowledge of the number of shape variations or the number of *thick* patterns in a database, we are required to perform unsupervised clustering.

2.2.1 Our clustering framework

We basically follow two steps [20,34] as explained below:

Step 1. Find the similarity or dissimilarity between every possible pair of *thick* patterns to create a distance (similarity or dissimilarity) matrix. This is achieved by representing a *thick* pattern p by a shape signature vector s_p of size i . and then choosing $\delta(s_a, s_b)$, the distance (similarity or dissimilarity) metric between the signatures of *thick* patterns a and b . The outcome will of course be highly dependent on both the choice of the shape descriptor and the metric used.

For instance, some of the more obvious metrics to construct the matrix are

1. *City-block* $\delta(s_a, s_b) = \|s_a - s_b\|_1 = \sum_i |s_a[i] - s_b[i]|$,
2. *Euclidean* $\delta(s_a, s_b) = \|s_a - s_b\|_2 = \sqrt{\sum_i (s_a[i] - s_b[i])^2}$ and
3. *Squared Euclidean* $\delta(s_a, s_b) = \|s_a - s_b\|_2^2 = \sum_i (s_a[i] - s_b[i])^2$.

Ways to choose and experimental validation of what combination of descriptor and metric give the best results are developed further in this paper.

Step 2. Group the similar *thick* patterns in the form of a hierarchical cluster tree.

In order to group them, we have again a choice of measuring overall distances (*linkage methods*) between clusters. We have chosen to implement three different types of linkage methods. All methods rely on the previously defined distance matrix.

1. Single-linkage clustering is sometimes known by *nearest neighbour* clustering. In this method, the distance between two clusters is computed as the distance between the two closest elements in two clusters. Mathematically, the distance between two clusters c_a and c_b can be expressed as,

$$D(c_a, c_b) = \min \{ \delta(s_a, s_b) : s_a \in c_a, s_b \in c_b \}. \quad (5)$$

2. Complete-linkage clustering uses the maximum distance between the two clusters. Mathematically, it can be expressed as,

$$D(c_a, c_b) = \max \{ \delta(s_a, s_b) : s_a \in c_a, s_b \in c_b \}. \quad (6)$$

3. Average-linkage clustering uses the mean distance between elements of each cluster.

$$D(c_a, c_b) = \frac{1}{|c_a| \times |c_b|} \sum_{s_a \in c_a} \sum_{s_b \in c_b} \delta(s_a, s_b). \quad (7)$$

Applied to our specific context, we consider the similarity matrix from all *thick* patterns. An agglomerative hierarchical clustering scheme consists in erasing rows and columns in this similarity matrix each time clusters are grouped together. While grouping, similarity is based on the chosen metric and linkage method. At every merge, we update the similarity matrix by deleting the rows and columns and replacing the merged distance values by the above linkage value. We repeat the process until all clusters are merged or it reaches a pre-set cluster-threshold. Figure 5 shows an example of a dendrogram using agglomerative hierarchical clustering. In this illustration, the clustering process ends up with a single cluster. The similarity between pairs is simply taken from one of the above linkage distance computations. For instance, clusters c_1 and c_2 are merged at a distance of 1.5. This is also called the *dendrographic distance*.

Until now, we have simplified the whole clustering process in two steps. However, there remain two unanswered questions.

1. Testing all possible combinations between both descriptor metrics and linkage methods will be time-consuming. Therefore, we shall try and compute an optimal combination via cluster verification.
2. What is the optimal number of clusters, or how and where to cut the cluster tree? Figure 5 only provides the clustering process until all clusters are merged, but not the cutoff threshold. To efficiently obtain the appropriate number of clusters, one either has to choose the threshold manually or set the best threshold by validating the clusters. Cluster validation requires either unsupervised or supervised approaches.

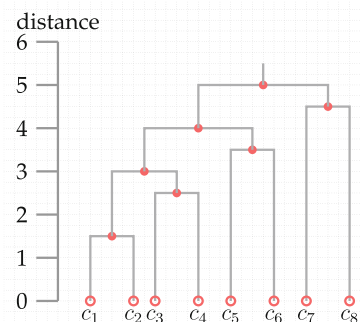
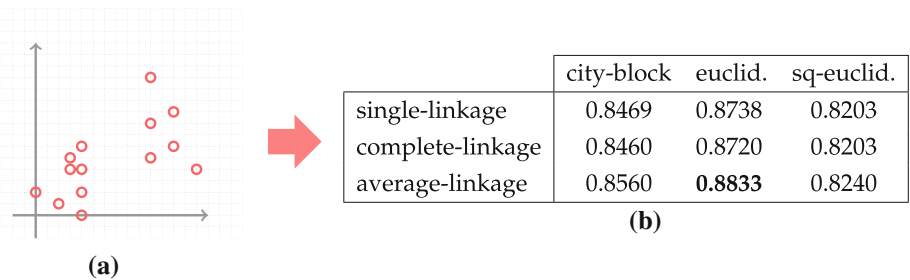


Fig. 5 Dendrogram example using eight *thick* patterns labelled with initial clusters c_1, c_2, \dots, c_8 . Pairs are merged based on their similarity

Fig. 6 Cluster verification. Cophenetic correlation coefficient from all possible combinations of distance metric and clustering linkage methods applied to a set of arbitrary features. The best combination is the one closest to 1. **a** Features. **b** Cophenetic correlation coefficient



Choices and experiments related to both cluster construction and validation will be developed in the next sections.

2.2.2 Cluster verification

Cluster analysis is highly sensitive to

1. shape descriptors,
2. distance metric selection and
3. linkage measure.

Approaches that use a different combination of distance metric and linkage method may yield different results (the choice of shape descriptors will be handled later). Furthermore, since the optimal combination depends on what shape signatures and data are being considered we need a dynamic approach to determine that optimum. Therefore, we are going to use the *cophenetic correlation coefficient* [9,35,46] for every combination and choose the best one before cluster validation.

In hierarchical clustering, the height of the link represents the distance between two clusters. This height is known as cophenetic distance.

Consider the original data $S = \{s_i\}$ having been clustered to produce a dendrogram Z . Let $\bar{\delta}$ be the average value of all distance measures $\delta(s_i, s_j)$ between the data samples and \bar{z} be the average of the $Z_{i,j}$, then the cophenetic correlation coefficient can be expressed as [9]

$$\text{Cop. Cor. Coeff.} = \frac{\sum_{i < j} (\delta(s_i, s_j) - \bar{\delta})(Z_{i,j} - \bar{z})}{\sqrt{\left[\sum_{i < j} (\delta(s_i, s_j) - \bar{\delta})^2\right] \left[\sum_{i < j} (Z_{i,j} - \bar{z})^2\right]}}$$

where $Z_{i,j}$ is the dendrogrammatic distance between the model points from the linkage function. The dendrogrammatic distance is the height of the node at which these two points are first joined together. Figure 5 gives an example.

The cophenetic correlation coefficient expresses a combined measure between these two different sets of values: One is from distance metric and another is from linkage function. Therefore, if the clustering is valid, the linking of patterns in a cluster tree should have a strong correlation with the distance between the clusters themselves. That is, the cophenetic correlation coefficient closer to 1 is the one

corresponding to the most accurately clustered patterns. To illustrate the idea, we take a set of arbitrary features to see how the cluster verification works and how we obtain the best combination of a distance metric and a linkage method. Figure 6 provides an example. We have provided the cophenetic correlation coefficient for all possible pairs of combinations. In this example, the combination of Euclidean distance metric and the average-linkage clustering method is found to be the best compared to others because the cophenetic correlation coefficient closer to 1. This means that we do not need to use remaining pairs for cluster validation. In Fig. 6, the number of combinations can be increased by using some more distance metrics. This illustration is only intended to be a general overview of how this concept works.

2.2.3 Cluster validation

In this part, we will be focussing on determining the correct number of clusters. The number of clusters has an influence on the overall aimed recognition performance. For example,

- if too many clusters are defined, they will be small in size and their elements (even inter-cluster) will be highly similar, but the analysis of many clusters increases time complexity and sensitivity to noise;
- if fewer clusters are defined, they will automatically be larger, and their elements will show less similarity to one another; they will be more robust to noise, but may smooth out significant shape details.

The evaluation measures that are applied to judge various aspects of cluster validity are traditionally classified into supervised and unsupervised approaches. In our case, since we do not have external input to fix the number of clusters, we use unsupervised techniques. Unsupervised measures of cluster validity are often based on internal indices: cluster cohesion and separation.

1. Cluster cohesion refers to compactness or tightness of the cluster. It expresses how closely related the objects in a cluster are.
2. Cluster separation refers to isolation of the clusters and how distinct or well separated a cluster is from other clusters.

ters. The clusters themselves should be widely separated. There are three common approaches measuring the distance between two different clusters

- the closest member of the clusters,
- the most distant members, and
- the centres of the clusters.

Within the framework, the indices we are going to use to validate clusters are as follows:

1. Dunn index (DU) [17]

It is defined as the ratio between the minimal intra-cluster distance to maximal inter-cluster distance. The Dunn index (DU) for k clusters is expressed as

$$DU_k = \min_{i=1,\dots,k} \left\{ \min_{j=i+1,\dots,k} \left(\frac{\text{dist.}(c_i, c_j)}{\max_{m=1,\dots,k} \text{dist.}(c_m)} \right) \right\}, \quad (8)$$

$$\text{and } \text{dist.}(c_i, c_j) = \min_{s_a \in c_i, s_b \in c_j} \delta(s_a, s_b) \text{ and}$$

$$\text{dist.}(c_m) = \max_{s_a, s_b \in c} \delta(s_a, s_b).$$

It tends to be maximal when the inter-cluster distances are large and the intra-cluster distances are small.

2. Davies–Bouldin index (DB) [14]

It identifies not only the clusters which are far from each other but also their compactness. The DB index is defined as

$$DB_k = \frac{1}{k} \sum_{i=1}^k \max_{j=1,\dots,k, i \neq j} \left\{ \frac{\text{dist.}(c_i) + \text{dist.}(c_j)}{\delta(c_i, c_j)} \right\}, \quad (9)$$

$$\text{and } \text{dist.}(c_i) = \frac{1}{n_i} \sum_{s_a \in c_i} \delta(s_a, s_i^{\text{mean}}),$$

where n_i is the number of elements and s_i^{mean} the centroid of cluster c_i . The DB index is minimal for the best number of clusters.

3. Silhouette index (SI) [36]

SI computes the silhouette width for each sample, average silhouette width for each cluster and overall average silhouette width for a total data set. The silhouette is based on the comparison of cluster tightness and separation. The average silhouette width is used to decide how good the number of selected clusters is. It is an average over all observations, i.e.,

$$SI_k = \frac{1}{n} \sum_{i=1}^n \frac{(\text{dist}\hat{2}_i - \text{dist}\hat{1}_i)}{\max(\text{dist}\hat{1}_i, \text{dist}\hat{2}_i)}, \quad (10)$$

where n is the total number of elements, $\text{dist}\hat{1}_i$ is the average distance between the element i and all other elements in its own cluster and $\text{dist}\hat{2}_i$ is the minimum of the average distance between i and elements in other clusters. SI is maximised for the best number of clusters.

4. Score function (SF) [39]

As in the DI and DB, SF is also based on inter-class and intra-class distance. It can be expressed as follows:

- Between class distance (bcd)

$$bcd = \frac{\sum_{i=1}^k \delta(s_i^{\text{mean}}, s_{\text{tot.}}^{\text{mean}}) \times n_i}{n \times k} \quad (11)$$

where k is the number of clusters of size n , s_i^{mean} the centroid of cluster c_i having n_i elements and $s_{\text{tot.}}^{\text{mean}}$ the centroid of all clusters.

- Within class distance (wcd)

$$wcd = \sum_{i=1}^k \left(\frac{1}{n_i} \sum_{s_a \in c_i} \delta(s_a, s_i^{\text{mean}}) \right). \quad (12)$$

Now the SF is computed as,

$$SF = 1 - \left(1/e^{bcd-wcd} \right). \quad (13)$$

The higher the value of the SF, the more appropriate the number of clusters, i.e., it maximises the bcd and minimises the wcd .

To attest the cluster validation indices, we take the example from Fig. 6 and try to deduce the best number of clusters. In Fig. 7 we can see that every cluster validation index converges at two clusters.

3 Symbol recognition

3.1 Graph matching principle

The recognition framework principally follows the corresponding relation alignment (presented in Santosh et al. [44,45]) for matching two graphs: $G1$ and $G2$, where $G(V, E, F_A, F_E)$. Then their distance is computed as

$$\text{Dist.}(G1, G2) = \sum_{r \in E} \delta(F1_E(r), F2_E(\sigma(r))), \quad (14)$$

where

- $\delta(,)$ computes the distance between two spatial relation signatures.
- $F1$ (resp. $F2$) is the function computing the spatial relation signature of an edge.
- $\sigma : E1 \rightarrow E2$ is the correspondence function mapping edges from one graph to the other. It has to be noted that, because of the unique fixed labelling of node attribute types, this mapping is immediate and exact, and there is no underlying matching difficulty.

Furthermore, as described in Sect. 2, a symbol \mathcal{S} has \mathcal{K} number of *thicks* and we have a set of $\{G_\kappa\}_{\kappa=1}^{\mathcal{K}}$ ARGs rep-



Fig. 7 Cluster validation (cf. Fig. 6). It is applied for all validation indices. For a number of clusters: {1, 2, 3, 4, 5 and 6} in **a**, every index produces two clusters. Visual illustration is provided on the *right side*. **a** Cluster validation. **b** Visual illustration

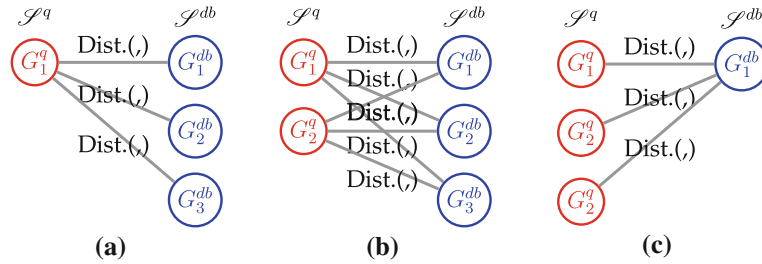


Fig. 8 A few examples showing possible graph matching scheme between two symbols \mathcal{S}^q and \mathcal{S}^{db} by providing various graph(s) that compose a symbol. In this illustration, each *encircled* token represents

a *graph*. This illustrates a basic idea without integrating *thick* pattern clustering. **a** 1×3 matching. **b** 2×3 matching. **c** 3×1 matching

representing it. Note that \mathcal{K} varies from one symbol to another. Now, we can represent a query symbol $\mathcal{S}^q = \{G_{\kappa}^q\}_{\kappa=1}^{\mathcal{K}}$ and a database symbol $\mathcal{S}^{db} = \{G_{\kappa'}^{db}\}_{\kappa'=1}^{\mathcal{K}'}$. To compute the similarity between the two symbols, the main idea is to find the best matched graphs pair. This means that we compute the distances between all possible pairs of graphs. That is, any database graph $G_{\kappa'}^{db}$ that is matched with any query graph G_{κ}^q is sufficient to find similarity between the symbols: \mathcal{S}^q and \mathcal{S}^{db} .

Let us take any pair of symbols \mathcal{S}^{\dagger} and \mathcal{S}^{\ddagger} to formally compute the minimum distance between them,

$$\Delta(\mathcal{S}^{\dagger}, \mathcal{S}^{\ddagger}) = \min_{\kappa} \left(\min_{\kappa'} \left(\text{Dist.} \left(G_{\kappa}^{\dagger}, G_{\kappa'}^{\ddagger} \right) \right) \right). \quad (15)$$

Figure 8 shows three possible cases to realise graph matching between them, without considering *thick* pattern clustering. Considering whole database symbols $\{\mathcal{S}^{db}\}_{db=1, \dots, \mathbb{DB}}$, the closest candidate for any query symbol \mathcal{S}^q , can be computed as, $\min_{db}(\Delta(\mathcal{S}^q, \mathcal{S}^{db}))$. Note that, we are not just looking for the closest candidate, but also retrieving database symbols for any provided short-list. In the following part, we explain about how retrieval can be handled.

3.2 An inclusion of *thick* pattern clustering in graph matching

The aforementioned mentioned matching concept suffers from heavy computation as soon as the number of *thick*

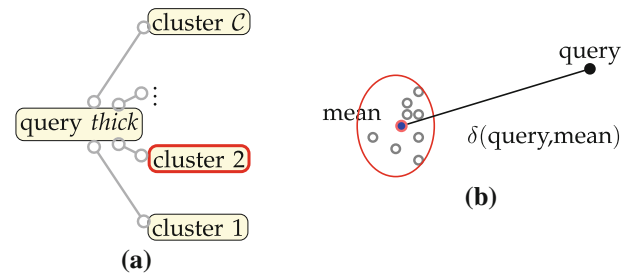


Fig. 9 Searching for the right cluster(s) where query *thick* pattern belongs to. In this illustration, the *red-boxed cluster* in **a** is the corresponding cluster for a query *thick*. To compute distance, only representative (mean) of every cluster will be taken into account as shown in **b**. **a** Searching for right cluster. **b** Computing distance between query *thick* and mean of the cluster (colour figure online)

increases. To recover this, we are going to use *thick* pattern clustering, as mentioned before.

Step 1. For each query symbol, the first step is to allocate the cluster in which the query *thick* belongs to. To compute the distance between any test *thick* pattern labelled with a shape signature, we use a centroid of the particular cluster, i.e., $\delta(s_a, s_i^{\text{mean}})$. It is shown in Fig. 9. More than one cluster can be selected if a query symbol has two or more *thick* patterns with different shape information. Figure 10 shows a couple of examples of it.

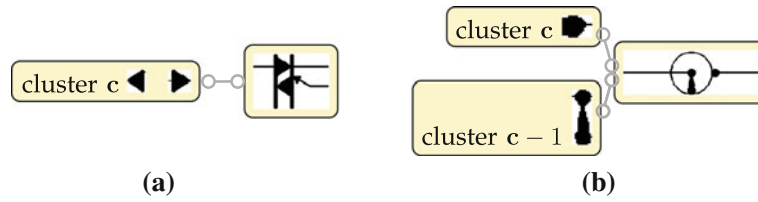


Fig. 10 Thick pattern(s) in cluster(s) and its (their) corresponding symbol. A symbol can have more than one cluster. It is primarily due to different shaped thick patterns. In **a**, both thick patterns appear in a single cluster, while there exist two different clusters in case **b**

Step 2. Once the cluster(s) is(are) selected, then the symbols related to those *thick* patterns, i.e., corresponding symbols are taken for matching. Our graph matching is explained in Eq. (14).

As a consequence, for matching, we have not taken

1. all database symbols; and
2. all graphs related to a particular database symbol.

This means that a limited number of database symbols are used, i.e., $\{\mathcal{S}^{db}\}_{db=1, \dots, \widehat{\mathbb{DB}}}$, where $\widehat{\mathbb{DB}} \subseteq \mathbb{DB}$. In addition, for each particular database symbol \mathcal{S}^{db} , the number of graph matchings can be reduced to $\hat{\mathcal{K}}' \subseteq \mathcal{K}'$. Clustering thus, helps to reduce running time.

3.3 Retrieval and ranking

The previously defined distance $\Delta(\cdot, \cdot)$ provides an idea of how similar/dissimilar a database symbol is, with respect to a query. In order for this similarity measure to fall into an appropriate range, we normalise $\Delta(\cdot, \cdot)$ to $[0, 1]$ by taking all filtered database symbols $\widehat{\mathbb{DB}}$,

$$\overline{\Delta}(\cdot, \cdot) = \frac{\Delta(\cdot, \cdot) - \Delta(\cdot, \cdot)_{\min.}}{\Delta(\cdot, \cdot)_{\max.} - \Delta(\cdot, \cdot)_{\min.}} \quad (16)$$

Now the matching score, $m.\text{score}(\mathcal{S}^q, \mathcal{S}^{db}) = 1 - \overline{\Delta}(\mathcal{S}^q, \mathcal{S}^{db})$. For any provided short-list, ranking can be made based on the decreasing order of matching score.

4 Experiments

In order to measure the impact of the vocabulary clustering compared to using relations only [44], we have used the same dataset and experimental protocol.

We use well-known state-of-the-art of shape descriptors used for *thick* pattern (vocabulary) clustering:

1. Zernike moments (ZM) [23],
2. \mathcal{R} -signature [47],
3. SC [4],
4. Generic fourier descriptors (GFD) [60] and
5. DTW-Radon (lets say \mathcal{D} -Radon) [43].

4.1 Dataset, ground-truth and evaluation metric

4.1.1 Dataset

Figure 11 gives an overview of the dataset we are using for our experiments. The global dataset is composed of roughly 500 different symbols, taken from electrical wiring diagrams [53]. It contains symbols that are either very similar in shape—and only differ by slight details—or, on the other hand, are completely different from a visual point of view. Symbols may also be composed of other known and significant symbols and need not necessary be connected.

4.1.2 Ground-truth and evaluation metric

Since there is no absolute ground-truth associated with our dataset, we have asked 6 volunteers to manually select what they consider as “similar” symbols, and we have merged their inputs to reduce possible subjective bias. They have chosen the candidates which have similar visual overall appearance or contain significantly similar parts with respect to the chosen query. In our testing protocol, we consider that a result returned from an algorithm is correct if at least one evaluator has selected the same result among similar items. In more formal terms, for each query, the “ground-truth” is considered to be the set of symbols formed by the union of all human-selected sets. In Fig. 11, we have provided a few samples of query symbols and the corresponding lists of relevant symbols.

For every query, we rank the symbols at the output based on the distance measure described in Sect. 3. Since the number of similar symbols, according to the ground-truth, may vary a lot from one query to another, we use retrieval efficiency [21] as a measure for retrieval quality. For a chosen query and for a fixed number of K returned symbols, it can be expressed as,

$$\eta_K = \begin{cases} n/N & \text{if } N \leq K \\ n/K & \text{otherwise,} \end{cases} \quad (17)$$

where n is the number of returned relevant symbols and N the total number of relevant symbols in the dataset. Note

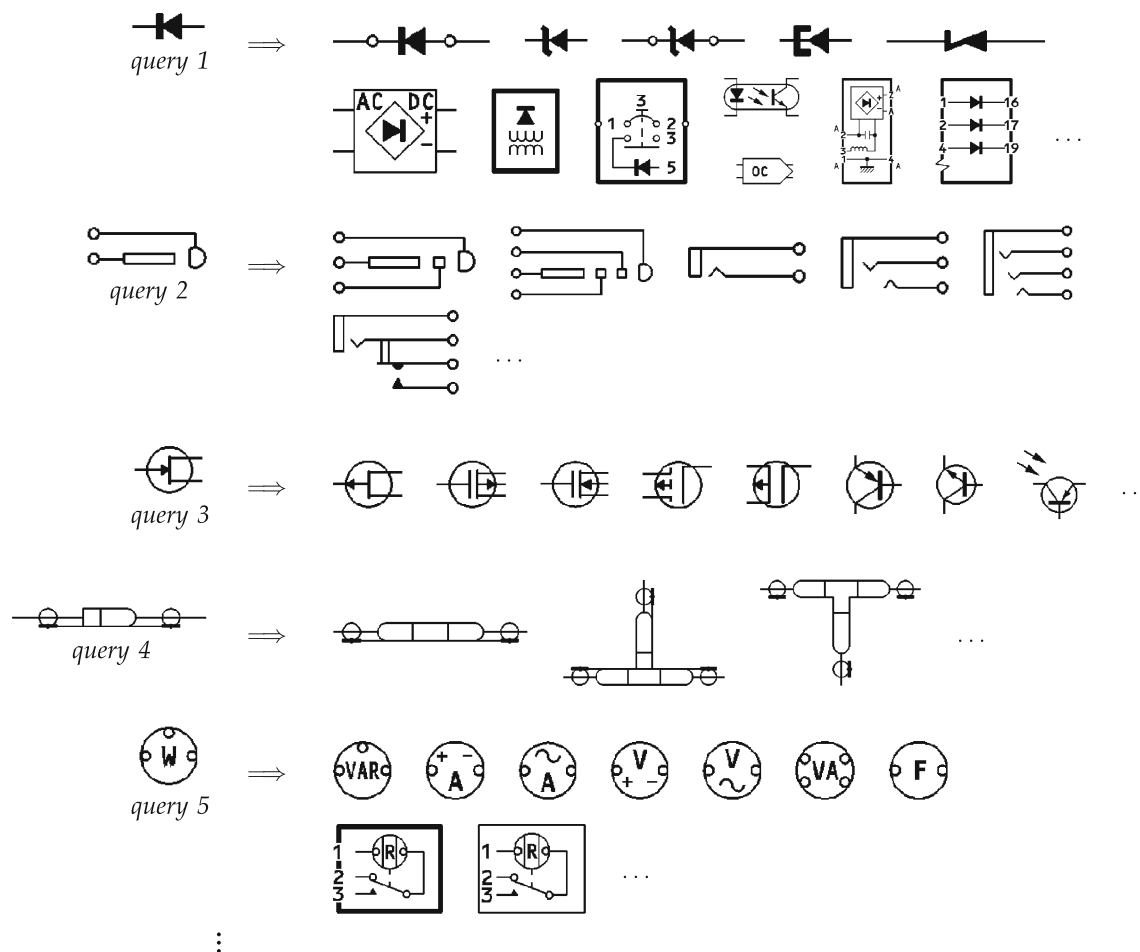


Fig. 11 A sample of electrical symbols. For every query symbol: *query 1* to *query 5*, a few human-selected relevant symbols are listed

that η_K computes the traditional recall if $N \leq K$ and computes precision otherwise. The main advantage of this is that the average retrieval efficiency curve is not biased even with different quantities of ground-truth for different queries, which happens when using the precision metric with $N < K$.

4.2 Thick pattern clustering

The goal of this section is to illustrate how choices of descriptors and cluster validation indices can have an influence on the obtained clusters. The way of actually selecting the best clustering parameters will be discussed in Sect. 4.3. We are considering the application of various shape signatures on *thick* patterns. Since discrimination power of shape descriptors vary from one to another, the number of clusters will obviously be different. Furthermore, a change in cluster validation indices also impacts the results, as we shall show here. We are going to test different shape descriptors and perform various clustering and cluster validation indices on each of

them in Sect. 4.3. The following section uses GFD [60] as an example.

4.2.1 Clustering using GFD

We have sampled the GFD parameters from 4 to 12 for the radial and from 6 to 20 for angular frequencies. For each of these values, we have computed the Dunn and DB clustering indices. Figure 12 shows the effect of these changes on the number of clusters obtained. For example, we get 33 clusters from Dunn and 30 clusters from DB when we take radial and angular frequencies values of (6, 16). For more clarity, each value in Fig. 12 (i.e., number of clusters in red box, for instance), we have provided a complete illustration in Fig. 13 about how cluster validation indices select different number of clusters. The observation is that depending on the chosen parameters (i.e., GFD parameters) and criteria (i.e., validation indices), the final number of clusters can vary widely.

Fig. 12 Number of clusters after cluster validation tests using various GFD parameters using Dunn and Davies–Bouldin indices. **a** Dunn index. **b** Davies–Bouldin index

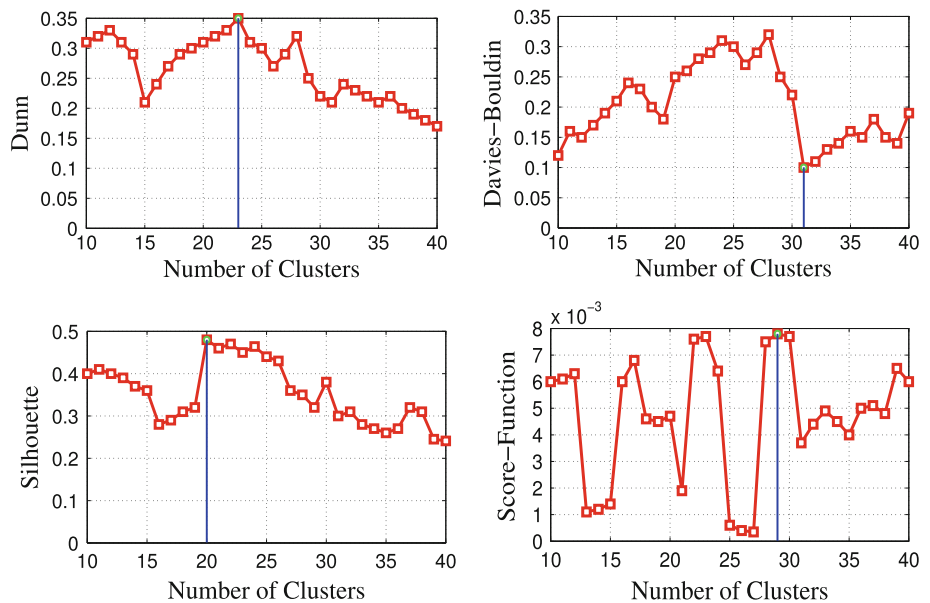
		Angular frequency									
		6	8	10	12	14	16	18	20		
Radial frequency	4	7	9	10	10	16	19	21	21	Radial frequency	4
	6	9	10	17	24	24	33	28	35		6
	8	9	9	21	23	33	38	38	40		8
	10	11	11	21	28	31	40	37	34		10
	12	21	27	31	33	38	33	30	28		12

(a)

		Angular frequency									
		6	8	10	12	14	16	18	20		
Radial frequency	4	9	12	11	12	19	21	22	24	Radial frequency	4
	6	10	13	19	25	22	30	29	33		6
	8	12	14	23	25	31	34	40	40		8
	10	9	11	21	25	32	38	32	35		10
	12	21	24	27	36	35	33	23			12

(b)

Fig. 13 An example of how cluster validation indices behave for GFD. The maximal value is shown by the vertical line in each plot



Using this framework, and for visual illustration, Fig. 14 shows a few clusters and their corresponding patterns from the DB cluster validation index.

4.2.2 Clustering using other shape descriptors

For the remaining shape descriptors, mentioned before, like SC, Zernike moments, \mathcal{R} -signature, \mathcal{D} -Radon, a similar cluster validation process has been carried out. In Sect. 4.3, the effect of the use of all shape descriptors on retrieval performance will be presented. This goes with the appropriate choice of cluster validation index.

4.3 Results and analysis

4.3.1 Related previous work

Since the approach is the extension of a previous work, we admit some of the points and conclusions described in Santosh et al. [44] and get directly to the point for comparisons with the existing state-of-the-art. For clarity, we recall some of the experimental setup choices, such as the use of retrieval

efficiency, as described in Sect. 4.1, which has been systematically computed for values of $K = 1$ to 10, over 30 queries, for all experiments described in this section.

1. In the previous approach, called RLM [44], we have studied the influence of different resolutions for the angular sampling steps. Its value represents the trade-off between the optimal choice of resolution—and thus precision of spatio-structural information capture—and time/space requirements. As reported in Santosh et al. [44], the best trade-off is 3° , which is the value we adopt for the rest of our experiments. During its comparison with most state-of-the-art spatial relation models such as cone-shaped [30], angle histogram [55] and MBR [31], MBR came out as the best performer among the three competing benchmarking relation models. RLM performed better than MBR with a substantial difference of more than 30 %.
2. Similarly, a series of experiments reported in Santosh et al. [43,44] study the appropriateness of a set of major state-of-the-art shape descriptors such as Zernike moments (ZM) [23], \mathcal{R} -signature [47], SC [4], GFD [60]

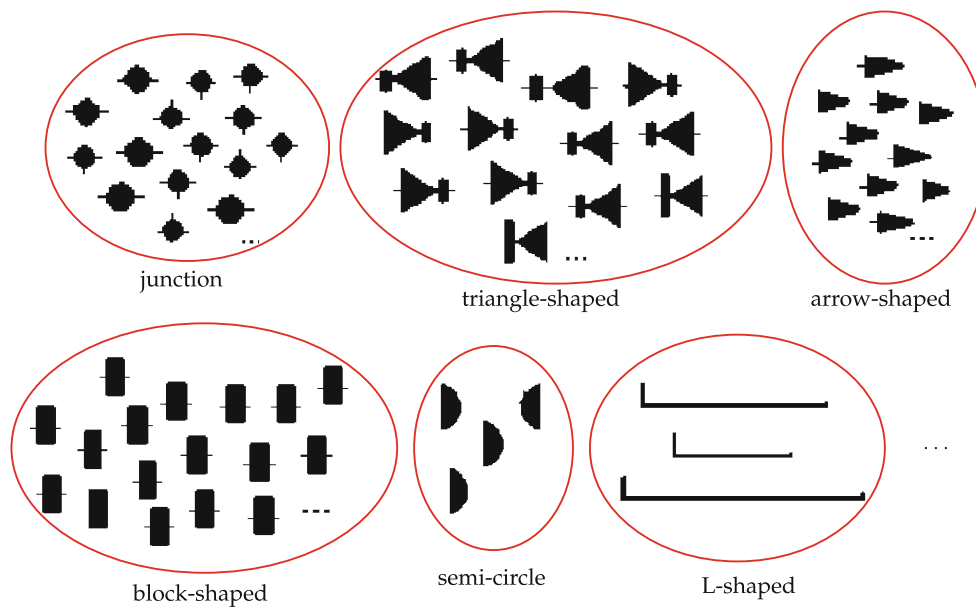


Fig. 14 A selection of pattern clustering results on *thick* components, using GFD shape descriptor and Davies–Bouldin

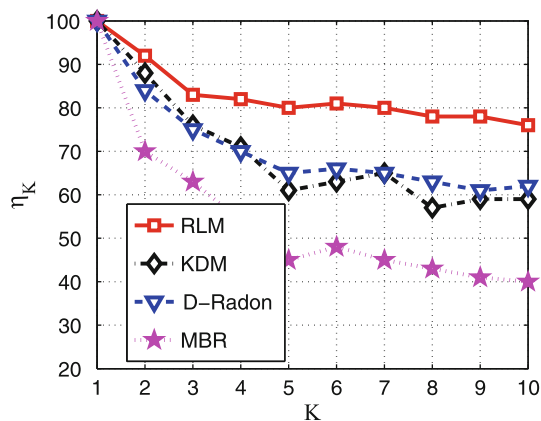


Fig. 15 Average retrieval efficiency with $K = 1$ to 10 over 30 queries

and \mathcal{D} -Radon [43]. \mathcal{D} -Radon performs best, but with a 12 % lower performance, on average, with respect to RLM.

- Furthermore, we have also compared RLM to pixel-based approaches specially designed for symbol recognition: statistical integration of histogram array (SIHA) [58] and Kernel density matching (KDM) [62], using the same protocol as described before. In this comparison, RLM outperforms KDM, the best performer of the two tested descriptors.

Figure 15 gives an overview of the observed performance differences between all tested methods.

4.3.2 Integrating vocabulary clustering with spatial relations

Following the reported results in Fig. 15, in this section, we aim to see how the integration of *thick* pattern selection (with spatial relations) improves the retrieval performance. Our query *thick* pattern selection is based on clustering mentioned in Sect. 2.2. Clustering performance is based on shape signatures and cluster validation indices. Therefore, we take both into account in order to assess a suitable combination for this application.

Figure 16 shows the comparison of performance of cluster validation indices for different shape descriptors. In the tests, we observe and analyse retrieval performances on a one-to-one basis. Overall, GFD outperforms all, but \mathcal{D} -Radon performs almost equally having a marginal difference. Zernike moments, shape context and \mathcal{R} -signature are lagging behind.


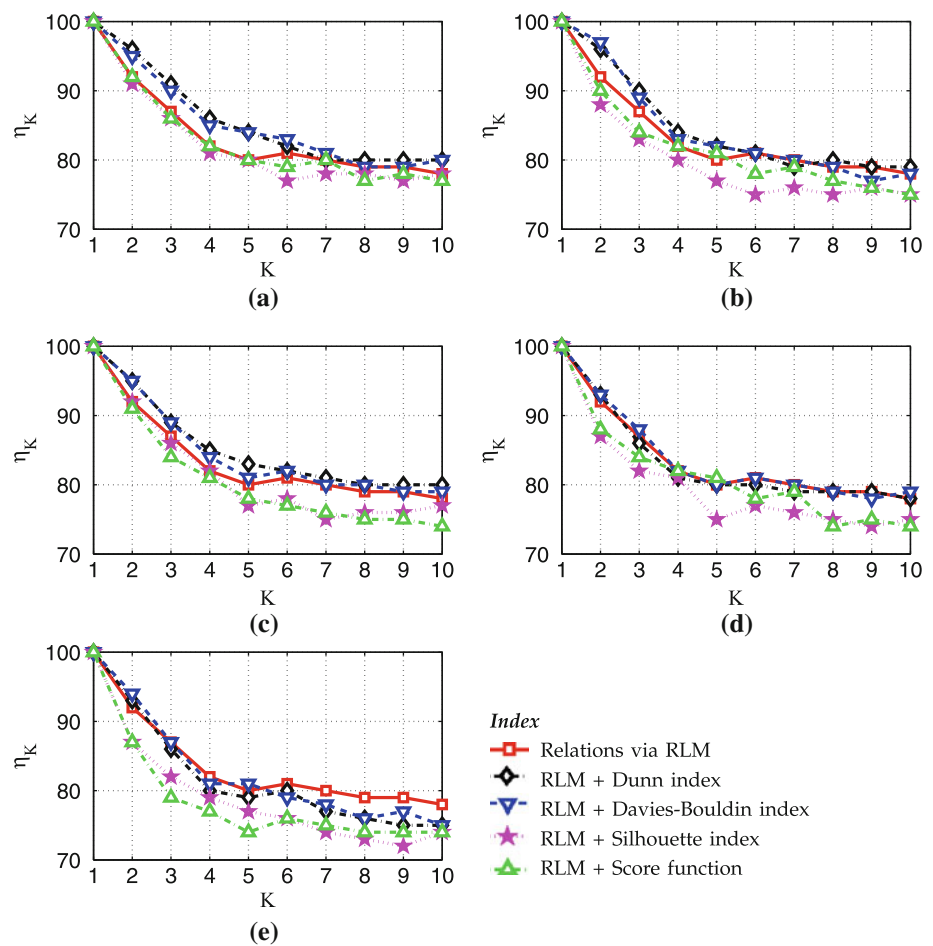

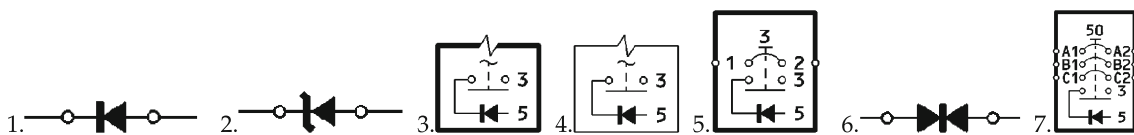
Selection of shape descriptors does not only provide a complete process, while it needs to account cluster validation indices, on the other hand. As said previously, different cluster validation indices provide different results. As a consequence, overall retrieval performance is affected. As shown in Fig. 16, for all shape descriptions, Dunn and DB indices provide almost similar advancements, while the remaining silhouette and score-function do not. Therefore, either Dunn or DB index will be the appropriate choice for this application. For a quick visual illustration, as an example  is taken as a query symbol. Using \mathcal{D} -Radon

Fig. 16 Average retrieval efficiency over requested list: 1–10 using global signal-based shape descriptors for *thick* patterns clustering and several different cluster validation indices. **a** GFD. **b** \mathcal{D} -Radon. **c** Shape context. **d** Zernike moments. **e** \mathcal{R} -signature



to describe the shape of *thick* , the first seven ranked database symbols retrieved are as follows,



A few more queries are illustrated in Fig. 17 where \mathcal{D} -Radon is used to describe the shape of the *thick* for clustering and DB index is used to validate the cluster. In this illustration, the first symbol on the top always corresponds to the query symbol and retrieved symbols are ranked from top to bottom based on the order of similarity. These retrieval examples provide an idea of how far the retrieval performance after integrating vocabulary clustering has been advanced. After integrating vocabulary clustering, we have found that retrieval symbols are visually better ranked in addition to its efficacy.

Along with relational signature matching, query *thick* pattern selection via clustering advances retrieval performance. However, no surprising difference is observed. The major

reason is that not all query symbols contain *thick* pattern in their vocabulary type sets. In other words, absence of *thick*

vocabulary type means ranking has been made only through relation alignment, with eventually no change in retrieval performance.

5 Conclusions

In this paper, we have presented the concept to describe the shape of the extracted visual vocabulary that shows significant shape variations and integrate relations that exist between them. Keeping the ARG framework, our intelligent concept of using shape descriptor via clustering provides an immediate retrieval applications. We have comprehensively studied unsupervised clustering and evaluated with several

	Relation			Vocabulary clustering + Relation		
	Q1	Q2	Q3	Q1	Q2	Q3
ranking ↓ 1.						
2.						
3.						
4.						
5.						
6.						
7.						
8.						
9.						
10.						

Fig. 17 Visual illustration of symbol ranking at the output for a few queries: $Q1$, $Q2$ and $Q3$, showing red tickmark for true retrieval and false, otherwise. The first symbol on the top always corresponds to the chosen query. Symbols are ranked from top to bottom based on decrease-

ing order of similarity. It uses \mathcal{D} -Radon to describe shape of the thick pattern in clustering and DB index to validate cluster (colour figure online)

different well-known validation indices. In our tests, we have observed the behaviour of the well-known state-of-the-art shape descriptors.

Very specifically, we have employed *thick* pattern clustering and use the related clusters with respect to the query for relation alignment. Clustering of thick patterns thus opens a global concept that it can be applied for any other visual primitives. Overall, we bring an attention to the use of a hybrid approach in symbol recognition since it combines both worlds: structural and statistical.

References

- Ah-Soon, C., Tombre, K.: Architectural symbol recognition using a network of constraints. *Pattern Recognit. Lett.* **2**, 231–248 (2001)
- Bar, M., Ullman, S.: Spatial context in recognition. *Perception* **25**, 324–352 (1993)
- Barrat, S., Tabbone, S.: A bayesian network for combining descriptors: application to symbol recognition. *Int. J. Doc. Anal. Recognit.* **13**(1), 65–75 (2010)
- Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(4), 509–522 (2002)
- Biederman, I.: Perceiving real-world scenes. *Science* **177**(43), 77–80 (1972)
- Bloch, I.: Fuzzy relative position between objects in image processing: new definition and properties based on a morphological approach. *Uncertain. Fuzziness Knowl. Based Syst.* **7**(2), 99–133 (1999)
- Bodic, P. L., Locteau, H., Adam, S., Héroux, P., Lecourtier, Y., Knippel, A.: Symbol detection using region adjacency graphs and integer linear programming. In: *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 1320–1324 (2009)
- Bunke, H. and Messmer, B. T.: Efficient attributed graph matching and its application to image analysis. In: Braccini, C. Floriani, L.D., Vernazza, G. (eds.) *Proceedings of the International Conference on Image Analysis and Processing*, vol. 974. *Lecture Notes in Computer Science*. Springer, Berlin, pp. 45–55 (1995)
- Carr, D.B., Young, C.J., Aster, R.C., Zhang, X.: *Cluster Analysis for CTBT Seismic Event Monitoring* (a study prepared for the US Department of Energy) (1999)
- Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. *Int. J. Pattern Recognit. Artif. Intell.* **18**(3), 265–298 (2004)
- Cordella, L.P., Vento, M.: Symbol and shape recognition. In: Chhabra, A.K., Dori, D. (eds.) *Graphics Recognition, Recent Advances*, vol. 1941. *Lecture Notes in Computer Science*, pp. 167–182. Springer, Berlin (2000a)
- Cordella, L.P., Vento, M.: Symbol recognition in documents: a collection of techniques? *Int. J. Doc. Anal. Recognit.* **3**(2), 73–88 (2000b)
- Coustaty, M., Bertet, K., Visani, M., Ogier, J.-M.: A new adaptive structural signature for symbol recognition by using a Galois lattice as a classifier. *IEEE Trans. Syst. Man. Cybern. Part B Cybern.* **41**(4), 1136–1148 (2011)
- Davies, D.L., Bouldin, D.W.: A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-1**(2), 224–227 (1979)

15. Dosch, P., Lladós, J.: Vectorial signatures for symbol discrimination. In: Lladós, J., Kwon, Y.-B. (eds.) *Graphics Recognition, Recent Advances and Perspectives*, vol. 3088. *Lecture Notes in Computer Science*, pp. 154–165. Springer, Berlin (2004)
16. Dosch, P., Tombre, K., Ah-Soon, C., Masini, G.: A complete system for the analysis of architectural drawings. *Int. J. Doc. Anal. Recognit.* **3**(2), 102–116 (2000)
17. Dunn, J.C.: Well-separated clusters and optimal fuzzy partitions. *J. Cybern.* **4**(1), 95–104 (1974)
18. Egenhofer, M., Herring, J.R.: *Categorizing Binary Topological Relations Between Regions, Lines, and Points in Geographic Databases*. Research report, University of Maine (1991)
19. Freeman, J.: The modelling of spatial relations. *Comput. Graph. Image Process.* **4**, 156–171 (1975)
20. Jain, A.K., Dubes, R.C.: *Algorithms for clustering data*. Prentice-Hall, Upper Saddle River (1988)
21. Kankanhalli, M.S., Mehre, B.M., Wu, R.K.: Cluster-based color matching for image retrieval. *Pattern Recognit.* **29**(4), 701–708 (1996)
22. Kara, L.B., Stahovich, T.F.: An image-based, trainable symbol recognizer for hand-drawn sketches. *Comput. Graph.* **29**(4), 501–517 (2005)
23. Kim, W.-Y., Kim, Y.-S.: A region-based shape descriptor using Zernike moments. *Signal Process. Image Commun.* **16**(1–2), 95–102 (2000)
24. Lamiroy, B., Guebbas, Y.: Robust and precise circular arc detection. In: Ogier, J.-M., Liu, W., Lladós, J. (eds.) *In: 8th International Workshop in Graphics Recognition. Achievements, Challenges, and Evolution (GREC)*, La Rochelle, France, July 22–23, 2009. *Selected Papers*, vol. 6020 of *Lecture Notes in Computer Science*. Springer, Berlin, pp. 49–60 (2010)
25. Lee, W., Kara, L.B., Stahovich, T.F.: An efficient graph-based recognizer for hand-drawn symbols. *Comput. Graph.* **31**(4), 554–567 (2007)
26. Lladós, J., Martí, E., Villanueva, J.J.: Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(10), 1137–1143 (2001)
27. Lladós, J., Valveny, E., Sánchez, G., Martí, E.: Symbol recognition: Current advances and perspectives. In: Blostein, D., Kwon, Y.-B. (eds.) *GREC—Algorithms and Applications*, vol. 2390. *Lecture Notes in Computer Science*, pp. 104–127. Springer, Berlin (2002)
28. Matsakis, P., Wendling, L.: A new way to represent the relative position between areal objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(7), 634–643 (1999)
29. Messmer, B.T., Bunke, H.: Efficient subgraph isomorphism detection: a decomposition approach. *IEEE Trans. Knowl. Data Eng.* **12**(2), 307–323 (2000)
30. Miyajima, K., Ralescu, A.: Spatial organization in 2D segmented images: representation and recognition of primitive spatial relations. *Fuzzy Sets Syst.* **2**(65), 225–236 (1994)
31. Papadias, D., Theodoridis, Y.: Spatial relations, minimum bounding rectangles, and spatial data structures. *Int. J. Geogr. Inf. Sci.* **11**(2), 111–138 (1997)
32. Pham, T.V., Smeulders, A.W.M.: Learning spatial relations in object recognition. *Pattern Recognit. Lett.* **27**(14), 1673–1684 (2006)
33. Rendek, J., Masini, G., Dosch, P., Tombre, K.: The search for genericity in graphics recognition applications: design issues of the qgar software system. In: Marinai, S., Dengel, A. (eds.) *Proceedings of the International Workshop on Document Analysis Systems*, vol. 3163. *Lecture Notes in Computer Science*. Springer, Berlin, pp. 366–377 (2004)
34. Reynolds, A.P., Richards, G., de la Iglesia, B., Rayward-Smith, V.J.: Clustering rules: a comparison of partitioning and hierarchical clustering algorithms. *J. Math. Model. Algorithm.* **5**(4), 475–504 (2006)
35. Rohlf, F.J., Fisher, D.R.: Tests for hierarchical structure in random data sets. *Syst. Biol.* **17**(4), 407–412 (1968)
36. Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**(1), 53–65 (1987)
37. Rusiñol, M. and Lladós, J.: Symbol spotting in technical drawings using vectorial signatures. In: Liu, W., Lladós, J. (eds.) *Graphics Recognition. Ten Years Review and Future Perspectives*, vol. 3926. *Lecture Notes in Computer Science*. Springer, Berlin, pp. 35–46 (2006)
38. Rusiñol, M., Borràs, A., Lladós, J.: Relational indexing of vectorial primitives for symbol spotting in line-drawing images. *Pattern Recognit. Lett.* **31**(3), 188–201 (2010)
39. Saitta, S., Raphael, B., Smith, I.F.: A bounded index for cluster validity. In: *Proceedings of the International Conference on Machine Learning and Data Mining in Pattern Recognition*. Springer, Berlin, pp. 174–187 (2007)
40. Salmon, J.P., Wendling, L., Tabbone, S.: Improving the recognition by integrating the combination of descriptors. *Int. J. Doc. Anal. Recognit.* **9**(1), 3–12 (2007)
41. Santosh, K.C., Lamiroy, B., Ropers, J.-P.: Inductive logic programming for symbol recognition. In: *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 1330–1334 (2009)
42. Santosh, K.C., Wendling, L., Lamiroy, B.: Unified pairwise spatial relations: an application to graphical symbol retrieval. In: Ogier, J.-M., Liu, W., Lladós, J. (eds.) *Graphics Recognition. Achievements, Challenges, and Evolution*, vol. 6020. *Lecture Notes in Computer Science*, pp. 163–174. Springer, Berlin (2010)
43. Santosh, K.C., Lamiroy, B., Wendling, L.: DTW for Matching Radon Features: A Pattern Recognition and Retrieval Method, vol. 6915. *Lecture Notes in Computer Science*. Springer, Berlin, pp. 249–260 (2011a)
44. Santosh, K.C., Lamiroy, B., Wendling, L.: Symbol recognition using spatial relations. *Pattern Recognit. Lett.* **33**(3), 331–341 (2011b)
45. Santosh, K.C., Lamiroy, B., Wendling, L.: Spatio-structural symbol description with statistical feature add-on. In: Kwon, Y.-B., Ogier, J.-M. (eds.) *Graphics Recognition. New Trends and Challenges*, vol. 7423. *Lecture Notes in Computer Science*, pp. 228–237. Springer, Berlin (2013)
46. Sokal, R.R., Rohlf, F.J.: The comparison of dendrograms by objective methods. *Taxon* **11**(2), 33–40 (1962)
47. Tabbone, S., Wendling, L., Salmon, J.-P.: A new shape descriptor defined on the Radon transform. *Comput. Vis. Image Understand.* **102**(1), 42–51 (2006)
48. Terrades, O.R., Valveny, E., Tabbone, S.: On the combination of ridgelets descriptors for symbol recognition. In: Liu, W., Lladós, J., Ogier, J.-M. (eds.) *Graphics Recognition. Recent Advances and New Opportunities*, vol. 5046. *Lecture Notes in Computer Science*, pp. 40–50. Springer, Berlin (2008)
49. Terrades, O.R., Valveny, E., Tabbone, S.: Optimal classifier fusion in a non-Bayesian probabilistic framework. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(9), 1630–1644 (2009)
50. Tombre, K.: Graphics recognition—what else? In: Ogier, J.-M., Liu, W., Lladós, J. (eds.) *Graphics Recognition. Achievements Challenges, and Evolution*, vol. 6020. *Lecture Notes in Computer Science*, pp. 272–277. Springer, Berlin (2010)
51. Tombre, K. and Lamiroy, B.: Pattern recognition methods for querying and browsing technical documentation. In: *Proceedings of the Iberoamerican Congress on Pattern Recognition*. Springer-Verlag, Berlin, pp. 504–518 (2008)
52. Tombre, K., Ah-Soon, C., Dosch, P., Habed, A., Masini, G.: Stable, robust and off-the-shelf methods for graphics recognition. In: *Proceedings of International Conference on Pattern Recognition*, vol. 1, p. 406 (1998)

53. Tooley, M., Wyatt, D.: Aircraft electrical and electronic systems: principles, operation and maintenance. In: Aircraft Engineering Principles and Practice. Electrical and Electronic Systems. Butterworth-Heinemann, Oxford (2008). ISBN:978-0750686952. <http://www.abebooks.com/9780750686952/Aircraft-Electrical-Electronic-Systems-Principles-0750686952/plp>
54. Valveny, E., Martí, E.: A model for image generation and symbol recognition through the deformation of lineal shapes. *Pattern Recognit. Lett.* **24**(15), 2857–2867 (2003)
55. Wang, X., Keller, J.: Human-based spatial relationship generalization through neural/fuzzy approaches. *Fuzzy Sets Syst.* **101**(1), 5–20 (1999)
56. Wenyin, L., Zhang, W., Yan, L.: An interactive example-driven approach to graphics recognition in engineering drawings. *Int. J. Doc. Anal. Recognit.* **9**(1), 13–29 (2007)
57. Xiaogang, X., Zhengxing, S., Binbin, P., Xiangyu, J., Wenyin, L.: An online composite graphics recognition approach based on matching of spatial relation graphs. *Int. J. Doc. Anal. Recognit.* **7**(1), 44–55 (2004)
58. Yang, S.: Symbol recognition via statistical integration of pixel-level constraint histograms: a new descriptor. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(2), 278–281 (2005)
59. Yuen, P.C., Feng, G.-C., Tang, Y.Y.: Printed chinese character similarity measurement using ring projection and distance transform. *Int. J. Pattern Recognit. Artif. Intell.* **12**(2), 209–221 (1998)
60. Zhang, D., Lu, G.: Shape-based image retrieval using generic Fourier descriptor. *Signal Process. Image Commun.* **17**(10), 825–848 (2002)
61. Zhang, D., Lu, G.: Review of shape representation and description techniques. *Pattern Recognit.* **37**(1), 1–19 (2004)
62. Zhang, W., Wenyin, L., Zhang, K.: Symbol recognition with kernel density matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(12), 2020–2024 (2006)