Symbol Recognition Using a Concept Lattice of Graphical Patterns

Marçal Rusiñol[1], Karell Bertet[2], Jean-Marc Ogier[2], and Josep Lladós[1]

[1] Computer Vision Center, Dept. Ciències de la Computació
Edifici O, UAB, 08193 Bellaterra, Spain
{marcal,josep}@cvc.uab.cat
[2] L3I, University of La Rochelle
Av. M. Crépeau, 17042 La Rochelle Cédex 1, France
{kbertet,jmogier}@univ-lr.fr

Abstract. In this paper we propose a new approach to recognize symbols by the use of a concept lattice. We propose to build a concept lattice in terms of graphical patterns. Each model symbol is decomposed in a set of composing graphical patterns taken as primitives. Each one of these primitives is described by boundary moment invariants. The obtained concept lattice relates which symbolic patterns compose a given graphical symbol. A Hasse diagram is derived from the context and is used to recognize symbols affected by noise. We present some preliminary results over a variation of the dataset of symbols from the *GREC 2005* symbol recognition contest.

Keywords: Graphics Recognition, Symbol Classification, Concept Lattices, Shape Descriptors.

1 Introduction

In order to tackle the problem of recognizing graphic symbols, a wide variety of symbol descriptors have been proposed in the literature. In most cases the applications have to cope with large corpora of graphical entities. In such conditions, the final performance of the systems not only depends on the description technique but also on which kind of data structure is used to provide efficient access and organize the feature descriptors.

In other cases, data structures are not used to provide efficient access to the data but also convey themselves some kind of information. In the field of Graphics Recognition the most clear example of such structures is the use of graphs, which have been applied over the years in structural pattern recognition problems. As other examples of data structures which have also been used in the symbol recognition domain due to its inherent codification of information we can cite for instance trees, dendrograms, or concept lattices.

Concept lattices are used as knowledge representation and its application to the symbol recognition domain was first proposed by Bertet and Ogier in [1]. The followed symbol recognition scheme by using concept lattices is to first build a

J.-M. Ogier, W. Liu, and J. Lladós (Eds.): GREC 2009, LNCS 6020, pp. 187–198, 2010.

binary table where the symbols correspond to the rows of the table and features from the descriptor vector correspond to the columns. A boolean value in a given cell indicates whether if a given symbol has a certain feature. From this table a Hasse diagram is derived and the classification of symbols is done in terms of traversal of this diagram. Guillas et al. presented in [5] a symbol recognition approach using concept lattices of pixel-based descriptors, whereas Coustaty et al. proposed in [3] the use of a structural description technique as features.

In this paper we propose a new approach to recognize symbols by the use of a concept lattice. Instead of using a numeric values arising from the feature vector, we propose to build the lattice in terms of symbolic patterns. Each symbol we want to recognize is represented by a set of composing primitives. Each one of these primitives is described by a well-known shape descriptor. The concept lattice relates which primitives compose a given graphical symbol. The obtained concept lattice from the context is then used to recognize symbols affected by noise. We present some preliminary results over a variation of the dataset of symbols from the *GREC 2005* Symbol Recognition Contest.

The remainder of this paper is structured as follows: the next section presents the followed steps to extract the primitives from the graphical symbols and how they are described. In section 3, we detail how the concept lattice is build from the set of model symbols and the sets of primitives. Section 4 presents the experimental setup by using a large dataset of distorted graphical symbols. Finally, the conclusions and future research lines can be found in Section 6.

2 Primitive Extraction and Symbol Description

Let us detail in this section how a graphical symbol is decomposed in a set of primitives representing simple graphical patterns, and how these primitives are described by the use of the well-known boundary moment invariants.

2.1 Extracting Primitives from Symbols

Our research work is mainly focused on the management of graphical data appearing in line-drawing images. Since these documents are mainly composed by lines, we choose to work with a vectorial representation of the symbols rather than at pixel level. In order to convert the symbol images to the vector domain, we use the raster-to-vector process proposed by Rosin and West in [9]. Instead of polygonally approximate the skeleton of the symbols, in our method, we approximate the contour of the closed loops conforming a symbol and its external contour in order to tackle with symbols which do not contain any loop at all. However, line segments are not suitable to be used as primitives due to its instability in terms of artifacts, fragmentation, errors in junctions, etc. A higher level entity has to be used as primitive. Adjacent vectors are merged together into a polyline instance. These polylines represent the graphical patterns conforming a given graphical symbol.

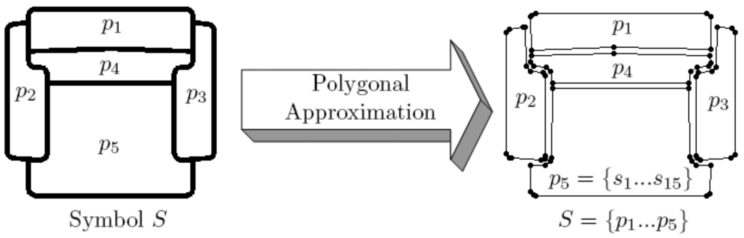


Fig. 1. Symbol representation in terms of a polygonal approximation of the contours of closed regions

Formally, let $p = \{s_1...s_n\}$ be a polyline consisting of n segments s_i. A symbol is represented in terms of its polylines representing loops and denoted as $S = \{p_1...p_m\}$. We can appreciate in Fig. 1 how the different parts of a symbol are detached making the regions meaningful primitives. Let us review in the next section how can we coarsely describe these primitives by the use of boundary moment invariants.

2.2 Primitive Description by Boundary Moment Invariants

The central $(p + q)$th order moment for a digital image $I(x, y)$ is expressed by

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y) \tag{1}$$

The use of the centroid $c = (\bar{x}, \bar{y})$ allows to be invariant to translation. The geometric moments can also be computed among the contour of the object as introduced by Chen in [2] and by Sardana et al. in [10] by using eq. 1 only for the pixels of the boundary of the object. A normalization by the object perimeter is used to achieve invariance to scale by using the following equation:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \quad \text{where} \quad \gamma = p + q + 1 \tag{2}$$

By sampling the polygonal approximation we can use the boundary moments as geometric descriptors of the primitives. In order to obtain invariance to rotation we use the set of seven functions proposed by Hu in [6] involving moments up to third order.

$$\begin{aligned}
\phi_1 = &\ \eta_{20} + \eta_{02} \\
\phi_2 = &\ (\eta_{20} - \eta_{02})^2 + (2\eta_{11})^2 \\
\phi_3 = &\ (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi_4 = &\ (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\phi_5 = &\ (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\
&\ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
\phi_6 = &\ (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 = &\ (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - \\
&\ (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
\end{aligned} \tag{3}$$

Moment invariants can be normalized to get the different invariants into similar numerical ranges. Hupkens and de Clippeleir proposed in [7] the following normalization of invariants to achieve a better robustness to noise.

$$\begin{array}{ll} \phi'_1 = \phi_1, & \phi'_4 = \phi_4 \;/\; \phi_1^3, \\ \phi'_2 = \phi_2 \;/\; \phi_1^2, & \phi'_5 = \phi_5 \;/\; \phi_1^6, \\ \phi'_3 = \phi_3 \;/\; \phi_1^3, & \phi'_6 = \phi_6 \;/\; \phi_1^4, \\ & \phi'_7 = \phi_7 \;/\; \phi_1^6 \end{array} \tag{4}$$

Formally, each primitive p_i of a symbol S is described by a seven-dimensional feature vector

$$f_i = [\phi'_1, \phi'_2, \phi'_3, \phi'_4, \phi'_5, \phi'_6, \phi'_7]$$

arising from the boundary moment invariant descriptors. The description space is quantized to transform this continuous set of values into a discrete set of symbolic graphical patterns. Let us detail in the next section how the concept lattice is build from the set of model symbols and the corresponding sets of primitives.

3 Concept Lattice of Graphical Patterns

Let us begin by reviewing the mathematical foundation of the concept lattices. We then focus on its application to the particular problem of symbol recognition by the traversal of the concept lattice.

3.1 Foundations of the Concept Lattice

We formally define a concept lattice by the formal concept analysis theory [4]. A concept lattice is a representation of a *formal context* $C = (G, M, R)$ where G is a set of *objects* and M is a set of *attributes*. R is a *relation* between these two sets. The fact that a certain object o has the attribute a is denoted as oRa.

From an object set $O \subseteq G$ we define as $f(O)$ the set of attributes in relation R with the objects from O.

$$f(O) = \{a \in M \mid oRa \; \forall o \in O\} \tag{5}$$

We analogously define $g(A)$ as being the set of objects in relation with the attributes from a set $A \subseteq M$.

$$g(A) = \{o \in G \mid oRa \; \forall a \in A\} \tag{6}$$

A *formal concept* for the context C is defined as a pair of objects and attributes (O, A) in relation according to R. The objects $O \subseteq G$ and the attributes $A \subseteq M$ must verify that $f(O) = A$ and $g(A) = O$. We denote as $\beta(C)$ all the concepts of the context C.

Formally, a concept lattice is defined as the set of concepts ordered by the *order relation*[1] $\leq$ defined for two concepts (O_1, A_1) and (O_2, A_2), as:

$$(O_1, A_1) \leq (O_2, A_2) \Longleftrightarrow O_1 \subseteq O_2 \tag{7}$$

the set of concepts and the order relation form the *concept lattice* $(\beta(C), \leq)$ of the context $C = (G, M, R)$.

By defining a *cover relation* $\prec$ as:

$$(O_1, A_1) \prec (O_2, A_2) \Longleftrightarrow \begin{cases} (O_1, A_1) < (O_2, A_2) \\ \nexists (O_3, A_3) \in \beta(C) \mid (O_1, A_1) < (O_3, A_3) < (O_2, A_2) \end{cases} \tag{8}$$

the *Hasse diagram* $(\beta(C), \prec)$ of a concept lattice $(\beta(C), \leq)$ is obtained.

A context may be seen as a table, where the objects correspond to the rows of the table and the attributes correspond to the columns. A boolean value in cell (o, a) indicates whether if a given object o has the attribute a.

Let us see in the next section how the concept lattices can be applied to the symbol recognition problem.

3.2 On the Use of Concept Lattices for Symbol Description

Concept lattices are used as knowledge representation and its application to the symbol recognition domain was first proposed by Bertet and Ogier in [1]. A classical symbol recognition scheme can define a context C where G corresponds to the set of graphical symbols we want to recognize and M corresponds to a set of attributes arising from the symbol descriptors. Guillas et al. presented in [5] a symbol recognition approach using concept lattices of pixel-based descriptors, whereas Coustaty et al. proposed in [3] the use of a structural description technique to define the context C.

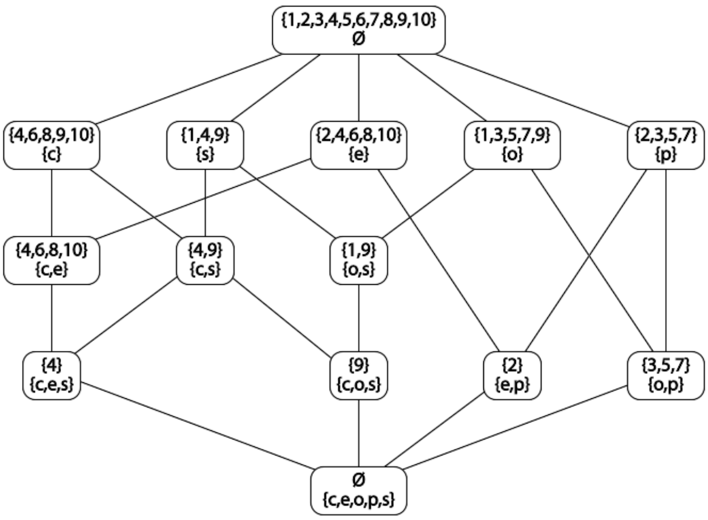
In all these previous approaches, graphical symbols are represented by a numerical descriptor. Each value of this feature vector is discretized in a number of intervals following a cutting criterion. Given a cutting value dividing a feature in two intervals, each model symbol has a membership relation with one of the intervals, that enables to differentiate the two subsets of symbols. The process of cutting the description space in intervals is repeated until each class can be distinguished. A concept lattice is build from the binary relationship between intervals and symbol families. When a symbol has to be recognized its description vector is also cut into intervals and the traversal of the Hasse diagram results in the class where the symbol belongs to.

However, these approaches may be very sensitive to noise, occlusions or even non-perfect symbol segmentations. If a single value of the feature vector is assigned to an incorrect interval, then the symbol can not be correctly recognized. In this paper we propose to describe symbols by graphical patterns taken as primitives and to build a concept lattice representing that a symbol family contain

[1] An order relation is a reflexive, antisymmetric and transitive binary relation.

Objects	**Attributes**				
	composite **c**	even **e**	odd **o**	prime **p**	square **s**
1			1		1
2		1		1	
3			1	1	
4	1	1			1
5			1	1	
6	1	1			
7			1	1	
8	1	1			
9	1		1		1
10	1	1			

(a) Context $C = (G, M, R)$



(b) Hasse diagram

Fig. 2. A concept lattice represented by a Hasse diagram for integers from 1 to 10 and several number attributes

or not a given simple shape. We can see an example of the proposed approach in Fig. 3. From the model symbols, we construct a set of attributes being graphical patterns. This set of attributes is constructed by clustering by similarity the space formed by all the feature vectors f_i describing the graphical primitives p_i composing the symbols in the database. The context C defines then a relationship between symbol classes and composing primitives. Let us see in the next section how we can use this context and the Hasse diagram derived from the concept lattice to recognize distorted symbols.

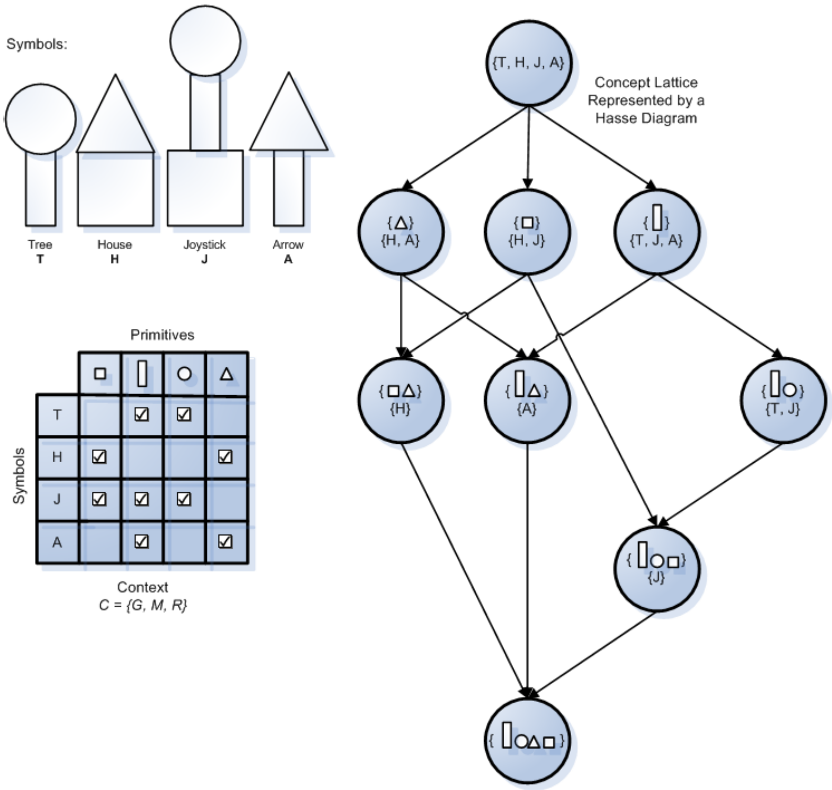


Fig. 3. Intuitive idea of the proposed approach

3.3 Traversing the Hasse Diagram for Symbol Recognition

Given a query symbol S^q and its corresponding feature vectors f_i^q describing each of the primitives p_i^q which compose the symbol, the Hasse diagram is traversed in order to recognize the query symbol. Starting from the topmost concept of the Hasse diagram, all the concepts of the poset containing a given set of attributes (primitives) of the query symbol are visited. A voting scheme accumulates evidences of the hypothetic symbols which may be the query symbol. These hypothetic symbols are the ones found in each poset concept.

Let us see the example in Fig 4. A noisy instance of a symbol from the family *joystick* has been taken as query symbol to recognize. The square and the circle primitives can be correctly identified despite the noise, however, the rectangle is not correctly recognized. When traversing the Hasse diagram, at each poset concept, we accumulate evidences of the plausible symbols. At the end, the symbol family accumulating more votes is taken as the class where the symbol belongs to.

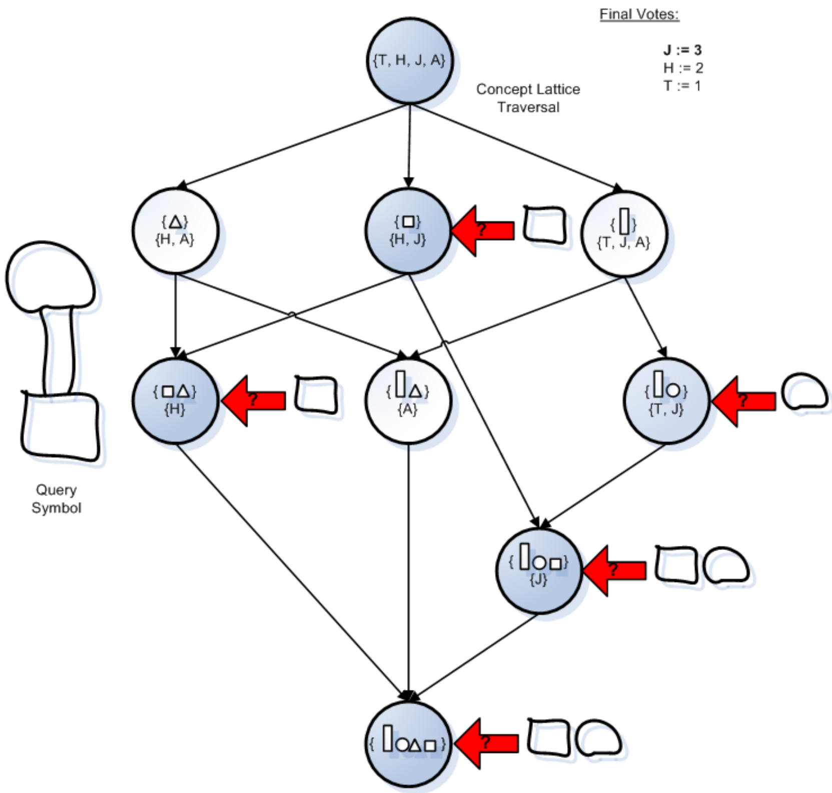


Fig. 4. Traversing the Hasse diagram for symbol recognition

In order to recognize symbols having different number of primitives a normalization of the voting space is done according to the theoretical number of votes which a given symbol should have obtained. A symbol having m primitives should receive $2^m - 1$ votes if all its primitives have been correctly identified. Let us see in the next section the obtained experimental results.

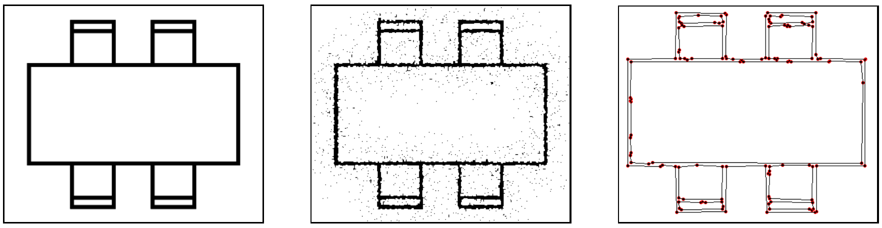
4 Experimental Results

We present in this section the experimental results for a symbol recognition problem. Let us first detail the symbol dataset we use and then present the obtained results.

4.1 Symbol Dataset

In order to carry out our experiments we have build a database of symbols in vectorial format with vectorial distortions. We have used all the 150 symbols of the original GREC2005 symbol database [11] as models. In order to generate realistic vectorial deformations, we first applied a degradation model to the bitmap images, and then applied a raster-to-vector process to these degraded images.

The bitmap images are degraded using the method presented by Kanungo et al. in [8] to simulate the noise introduced by the scanning process. Three different parameter configurations are used to obtain three different degradation levels. Some simple morphological operations are applied to these degraded images to get rid of the background noise. A connected component analysis is applied to label the closed regions and to extract the internal and external contours composing a symbol. These distorted contours are then polygonally approximated by using the Rosin and West algorithm introduced in [9]. In this dataset, the graphical symbols are composed by several polylines each one composed by a set of adjacent segments. The number of polylines which composes a symbol is constant for a given class, but the number of segments of these polylines is affected by the distortion model and varies from an instance to another. Fig. 5 shows an example of this distortion as well as some complementary characteristics of this dataset.[2]



(a) Bitmap model. (b) Degraded symbol with Kanungo noise (level 2 of 3). (c) Distorted symbol.

Property	**Value**
Number of classes	150
Total number of elements	45,000 (300 elements/class)
Max. number of polylines in a symbol	16
Min. number of polylines in a symbol	1
Mean number of polylines in a symbol	3.9
Max. number of segments in a symbol	264
Min. number of segments in a symbol	11
Mean number of segments in a symbol	73.7

(d) Details on the GREC-POLY database.

Fig. 5. Example and characteristics for the GREC-POLY database

[2] The vectorial symbol dataset is public available and can be downloaded through the following website http://www.cvc.uab.cat/∼marcal/GREC-POLY/

4.2 Evaluation

We present in Table 1 the obtained recognition results for the whole recognition experiment. Each one of the 45,000 degraded symbols is classified into the most likely symbol class depending on the value of the votes. However, we show in Table 1 the obtained recognition rates when considering just the topmost element, the two highest classes or the first three classes.

As we can appreciate, the noise introduced by the lowest distortion level is quite well tolerated. However, the deformation of the medium and higher level really impairs the overall performance of the method. Nevertheless, the box plot shown in Fig. 6, indicates that the performance is also highly dependent on the symbol design. Even in the highest level of distortion, the upper quartile attains good recognition rates whereas in the lowest level of distortion some symbol designs are badly recognized provoking some outliers in the box plot.

Table 1. Recognition results

Considered Results	**Recognition rates**(%)		
	Distortion levels		
	1 (low)	2 (medium)	3 (high)
top 1	78.93	64.61	53.92
top 2	80.41	66.03	55.24
top 3	80.58	66.13	55.38

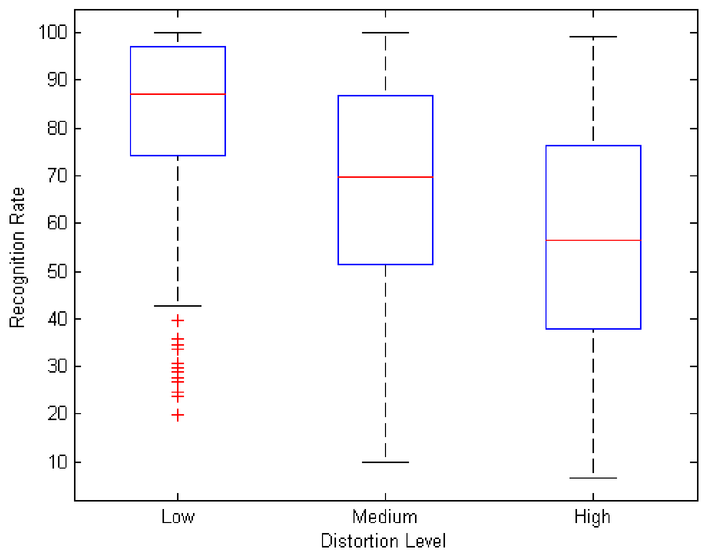


Fig. 6. Box plot of the recognition rates depending on the distortion levels

5 Conclusions and Future Work

In this paper we have proposed a new approach to recognize symbols by the use of a concept lattice of graphical patterns. Each model symbol has been decomposed in a set of graphical patterns taken as primitives. Each one of these primitives has been described by boundary moment invariants. The obtained concept lattice relates which symbolic patterns compose a given graphical symbol. The concept lattice was then used to recognize symbols affected by noise. We have presented some preliminary results over a variation of the dataset of symbols from the *GREC 2005* symbol recognition contest.

Despite the simplicity of the used descriptor, the obtained results are encouraging. The use of concept lattices as knowledge representation and its combination with voting approaches accumulating evidences to validate symbol class hypotheses seems a promising approach. The main novelty of this paper is the use of concept lattices from a symbolic set of attributes instead of numeric ones used in the previous approaches. The use of symbolic description of graphical symbols has been proven to be a powerful tool. We believe that this research line has to be further studied since there is still room for improvements.

The remaining challenge is to try to apply such kind of approaches to recognize non-segmented graphical symbols which may appear within its real context. The combined use of the knowledge representation given by the concept lattices and some spatial coherence rules may be envisaged in order to tackle with the problem of recognizing graphical symbols appearing in complete documents.

Acknowledgments

This work has been partially supported by the Spanish projects TIN2006-15694-C02-02, TIN2009-14633-C03-03 and CONSOLIDER-INGENIO 2010 (CSD2007-00018).

References

1. Bertet, K., Ogier, J.M.: Graphic Recognition: The Concept Lattice Approach. In: Lladós, J., Kwon, Y.-B. (eds.) GREC 2003. LNCS, vol. 3088, pp. 265–278. Springer, Heidelberg (2004)
2. Chen, C.C.: Improved Moment Invariants for Shape Discrimination. Pattern Recognition 26(5), 683–686 (1993)
3. Coustaty, M., Guillas, S., Visani, M., Bertet, K., Ogier, J.M.: On the Joint Use of a Structural Signature and a Galois Lattice Classifier for Symbol Recognition. In: Liu, W., Lladós, J., Ogier, J.-M. (eds.) GREC 2007. LNCS, vol. 5046, pp. 61–70. Springer, Heidelberg (2008)
4. Ganter, B., Wille, R.: Formal Concept Analysis – Mathematical Foundations. Springer, Heidelberg (1997)
5. Guillas, S., Bertet, K., Ogier, J.M.: A Generic Description of the Concept Lattices' Classifier: Application to Symbol Recognition. In: Liu, W., Lladós, J. (eds.) GREC 2005. LNCS, vol. 3926, pp. 47–60. Springer, Heidelberg (2006)

6. Hu, M.: Visual Pattern Recognition by Moment Invariants. IRE Transactions on Information Theory 8, 179–187 (1962)
7. Hupkens, T.M., de Clippeleir, J.: Noise and intensity invariant moments. Pattern Recognition Letters 16(4), 371–376 (1995)
8. Kanungo, T., Haralick, R.M., Philips, I.: Global and Local Document Degradation Models. In: Proceedings of the Second International Conference on Document Analysis and Recognition, ICDAR 1993, pp. 730–734 (1993)
9. Rosin, P.L., West, G.A.W.: Segmentation of Edges into Lines and Arcs. Image and Vision Computing 7(2), 109–114 (1989)
10. Sardana, H.K., Daemi, M.F., Ibrahim, M.K.: Global Description of Edge Patterns Using Moments. Pattern Recognition 27(1), 109–118 (1994)
11. Valveny, E., Dosch, P.: Report on the Second Symbol Recognition Contest. In: Liu, W., Lladós, J. (eds.) GREC 2005. LNCS, vol. 3926, pp. 381–397. Springer, Heidelberg (2006)