Performance Evaluation of Symbol Recognition

Ernest Valveny[1] and Philippe Dosch[2]

[1] Centre de Visió per Computador, Edifici O, Campus UAB,
08193 Bellaterra (Cerdanyola), Barcelona, Spain
ernest@cvc.uab.es

[2] Loria, 615, rue du jardin botanique,
B.P. 101, 54602 Villers-lès-Nancy Cedex, France
Philippe.Dosch@loria.fr

Abstract. Symbol recognition is one of the central problems in the field of graphics recognition. Many methods and approaches have been developed in the context of several application domains. In the last years, the need for generic methods, able to perform well on large sets of symbols in different domains, has become clear. Thus, standard evaluation datasets and protocols have to be built up in order to be able to evaluate the performance of all these methods. In this paper we discuss several points which should be taken into account in the design of such evaluation framework, raising a number of open questions for further discussion. These issues were the starting point of the organization of the contest on symbol recognition held during the last Workshop on Graphics Recognition (GREC'03). We also summarize the main features of the dataset and the protocol of evaluation used in the contest, as a first step to define a general evaluation framework, giving answer to these open questions.

1 Introduction

Performance evaluation has become an important field of interest among computer vision researchers. As more methods are available to solve a particular problem, there is a need to evaluate and compare them in order to determine strong and weak points for each of them. Thus, image databases, performance metrics and performance evaluation protocols have been developed in several areas of computer vision. In document analysis, some examples can be found in the performance evaluation of thinning algorithms [1] and OCR systems [2].

In the last years, the graphics recognition community has also become aware of the importance of evaluation and several contests concerning different graphics recognition problems have been organized using the framework of the International Workshop on Graphics Recognition (GREC). These contests have focused, up to now, on the raster-to-vector conversion of line drawings, specifically on the detection of dashed-lines [3], the general vectorization problem [4, 5] and the detection of arcs [6]. As a result of this effort, several metrics and protocols for the evaluation of line detection algorithms have been developed [7–9].

Another important subject in the development of any graphics recognition system is symbol recognition. In the conclusions of the past editions of the GREC

S. Marinai and A. Dengel (Eds.): DAS 2004, LNCS 3163, pp. 354–365, 2004.

Workshop, it has been pointed out the need for more generic symbol recognition systems and so, the need for a framework to evaluate them. The first attempt to address this issue was carried out in the more general framework of the 15th International Conference on Pattern Recognition (ICPR'00) [10], where a contest was organized using a limited set containing 25 symbols from one single domain, electronics. These symbols were scaled and degraded with a small amount of binary noise to generate the test data. A more general approach to the evaluation of generic symbol recognition systems was attempted in the last edition of the GREC Workshop [11], where 50 symbols from the domains of electronics and architecture were considered. Models of binary degradation and vectorial distortion were applied to the data to generate noisy images.

These contests have to be taken as the first steps in the definition of a general framework for performance evaluation of symbol recognition methods. In fact, the evaluation of symbol recognition is not a straightforward task. As we will show in the next sections, symbol recognition is a research field covering a lot of aspects about the application domain, the kind and the format of the data, the noise and the degradation of images, etc. Moreover, there are a lot of specific methods and approaches which are only capable to deal with some of these characteristics. Some reviews about the state-of-the-art in symbol recognition can be found in [12–14]. A general framework for symbol recognition evaluation should take into account all these factors, and should be flexible enough in order to be applied to most methods and approaches.

In general, a framework for performance evaluation must include the data set and the ground-truth, the definition of some performance metrics, and the description of a protocol to run a method on the data set and compare the results with the ground-truth. When trying to define such a framework for the evaluation of symbol recognition, a lot of open questions arise due to the complexity and diversity of the symbol recognition problem. In this paper we review these open questions, according to our experience in the organization of the symbol recognition contest held during GREC'03. We begin, in section 2, with a review of the main characteristics of symbol recognition, this description being essential to understand what are the main issues for a performance evaluation in this domain. Then, in section 3 we focus on the questions concerning the data set and the ground-truth. Performance metrics are discussed in section 4 and the framework that has to be built up for evaluation in section 5. Finally, in section 6 we explain the framework defined for the first edition of the symbol recognition contest. This framework can be seen as a first step to give answer to the questions discussed in previous sections. In section 7 we state some conclusions and draw several issues to work on in order to reach a general framework.

2 Characteristics of Symbol Recognition

2.1 Domain

Symbol recognition is a research topic related to a lot of application domains (architecture, electronics, mechanics, etc.) The symbol, whatever the type of a document, expresses in a compact way a known element that would be time-consuming and complex to represent each time it appears on a document. Thus, a symbol can be defined as a graphical entity with a particular meaning in the context of a specific application domain. As a result, the understanding of a document is possible only if the symbols it contains are known.

The representation of a symbol may include some graphical primitives, such as segments, arcs of circle, solid regions, solid and dashed strokes... but also in some domains, some arbitrary shapes, textures or color. As noticed in [14], symbols can be composed of different subsets of these graphical primitives, depending on the application domain, leading to different visual properties and configurations. Some examples of this diversity are presented in figure 1. This large variety of data has led to the design of a large number of recognition methods, based on different approaches, including some *ad-hoc* methods, specifically designed to recognize a family of symbols, sometimes in a particular context.

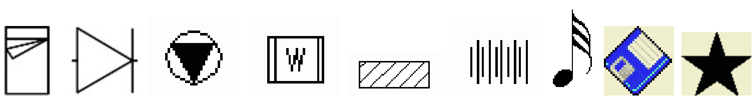


Fig. 1. Some examples of various symbols of different application domains.

2.2 Data Representation

Whatever the application domain, data can be available in several formats, depending on the nature of the original source. In symbol recognition, two representations are mainly handled: bitmap images and vectorial images. Bitmap images, usually produced by original document scanning, may be encoded using several color models: binary (black and white), grey-level or color. Vectorial images, produced by CAD softwares or after applying a vectorization step on bitmap images (see below), are represented using graphical primitives (arcs, segments, ...) and attributes (color, thickness, stroke style...)

Data representation often determines a kind of recognition approach. Thus, a bitmap document, carrying semantically poor information, is usually analyzed by statistical methods. A vectorial image, carrying more symbolic information, leads to “high-level” recognition methods, generally based on structural and syntactic approaches. A vectorization step allows to convert a bitmap image into a vectorial description, and then to apply “high-level” recognition methods on bitmap images. But this approach raises new issues, as vectorization itself is a critic process in document analysis. It has been the subject of past contests [4, 5], and even if it has been widely studied [15] and could overall be considered as a “mature” process, there is still room for improvement. In particular, vectorization still leads to typical artifacts, as inaccurate location of junctions, small segments... Moreover, different vectorization methods yield different results, and some of them can be better suited for a given recognition method.

2.3 Segmentation

A graphical entity is known as *segmented* whenever all the graphical parts of it are plainly identifiable. In the case of symbols, previous segmentation makes easier the recognition task, as the location of the symbol is known and the amount of data to be considered is reduced. But in a lot of documents, symbols are not segmented, and it is not easy to apply a previous generic segmentation process. In fact, in these documents, recognition sometimes raises a paradox: to be able to recognize symbols, they have to be segmented, and to be able to segment, symbols have to be recognized. Thus, recognition methods have to identify symbols as well as other graphical parts of the document (the graphical environment).

Segmentation ability is influenced by several factors. On the one hand, symbol segmentation is domain sensitive. Whenever the graphical environment can be easily modelled, symbols can be easily segmented. This is the case for musical scores for example, and more generally for symbols that are not connected to other graphical parts of a document. On the other hand, symbol segmentation is also approach sensitive. Some recognition methods are well-known to be NP-complex when working on non-segmented data. Other approaches, like symbol signatures, are able to determine areas of interest, that are likely to contain symbols. Thus, they can be taken as a kind of segmentation which facilitates the recognition task. Clearly, one of the main issues in symbol recognition is the ability for a method, or a chain of methods, to segment. This still remains an open issue, as noticed in [14].

2.4 Degradation of Data

Whatever the domain, the kind of acquisition or exploitation format, and the kind of symbols, data are often degraded. There are many different types of degradation. Sometimes, the original document is degraded itself, as an effect of photocopying, hand-drawing, rotation, scaling, resolution, manipulation... But degradation also occurs during the analysis of the documents: scanning, binarization, thinning, vectorization, etc. are well-known to produce artifacts. Some recognition methods work well on clean data, but are unable to handle some of the degradation types presented above. See [16] for a good presentation of the problem and [17] for an implementation of an automatic degradation method.

3 Data for Symbol Recognition Evaluation

3.1 Principles

According to the general description of symbol recognition in section 2, several questions arise when trying to define the data to be used in the evaluation. Maybe, the first question concerns the application domain of the data. We have seen that symbol recognition covers different domains, that symbols in each domain can be composed of different primitives, and that each domain can have specific constraints for recognition. So, as a first conclusion, a general framework for performance evaluation should take into account this diversity of data by including a representative sample of existing symbols. Indeed, no recognition method should be favored by the choice of the data, and the diversity is essential to assess the generic recognition potential of each method. Clearly, the more general the data set is (symbols mixed from all domains), the more we favor the development of general recognition methods. However, we can prevent some specific methods to take part in the evaluation, as some methods are only able to work on some specific domains and kind of data. One possible solution could be the definition of several data sets, one for each different domain or group of primitives. Then, we could asses the performance of a method for each independent set, but we could also combine performance on all sets to assess the generic recognition ability.

A major division of symbol recognition methods is on which data format (bitmap or vectorial) they rely. The main problem concerns vectorial data as the method employed for vectorization can influence the results of recognition. Then, we should wonder if it is better to provide a common vectorization tool to be used for all recognition methods or if we should let each method use its own vectorization approach. In the first case, if we provide a common vectorization tool, we can favor the methods best adapted to that vectorization approach. In the last case, we are not evaluating only recognition, but vectorization plus recognition. In both cases, results could not be comparable. So another idea is to base the data set exclusively on vectorial format, and to generate bitmap images from the vectorial representation. This approach presents an obvious interest for performance evaluation. This kind of conversion is more conservative of the intrinsic properties of an image than the reverse operation, as the visual representation of a vectorial image and that of its conversion into bitmap format are the same. The ground-truth is also easier to build and recognition methods can use both formats.

Segmentation is another important issue concerning the evaluation of symbol recognition. The question is whether to use real drawings including non segmented symbols, or if it is better to use only pre-segmented images containing only one symbol. As with vectorization, in the last case we only focus on the evaluation of recognition and not on the evaluation of segmentation plus recognition. But, as there is not any generic method for segmentation and as real applications of symbol recognition must usually include segmentation too, it is also interesting to consider real drawings with non segmented symbols from different domains. In this case, the generation of the ground-truth becomes more complex and performance measures should take into account issues such as missing symbols and false detections, the accuracy of location, and the accuracy of recognition itself.

Finally, we should include in the data set images with the most usual kinds of noise and degradation in real drawings. Obviously those images should be highly representative of all possible sources of noise. However, as there are many kinds of degradation, acquiring and annotating such a set from real images can imply a huge manual effort. Some alternatives are discussed in next section.

3.2 Generation of Data and Ground-Truth

The generation of the data set to be used in any performance evaluation process requires two fundamental tasks. Firstly, we need to collect a high number of images, representative enough of all domains and including all kinds of primitives and noise. Indeed, in order to achieve really significant measures, a lot of images are needed, from a lot of different sources and acquired under several acquisition conditions and constraints. All possible sources of noise and degradation should be taken into account: scanning at different levels of resolution, photocopying, printing at low resolution, old and degraded documents, hand-drawn documents, etc. Secondly, we need to annotate all these images in order to generate the ground-truth to be used in assessing the results of the methods.

There are two different approaches to get the data set and the ground-truth: the use of real data or the generation of synthetic data. Clearly, real data allows to evaluate symbol recognition methods with an exact replica of what is to be found in real applications. Therefore, performance measures obtained using real data are good estimators of performance in real drawings. However, the use of real data has some disadvantages concerning the acquisition, the organization of the data, and the generation of the ground-truth. Indeed, the generation of the ground-truth for all these documents implies a lot of manual effort, also needed to get segmented images of symbols from real drawings. An alternative way to get the data set and its ground-truth, specially for segmented images, is the use of synthetic data, *i.e.* the definition of one or several methods to automatically generate a great deal of images. With this approach, the main problem is the definition of suitable models for the automatic generation of images resembling the usual kind of noise and variations found in real images: scanning noise, degradation of old documents, variability produced by hand-drawing, etc. And the ground-truth is easier to set up, as it is implicitly available.

4 The Evaluation of Symbol Recognition

Evaluation could be viewed as a global measure allowing to determine the “best” recognition method. In fact, for previous contests on vectorization or arc recognition, some metrics have been defined with more or less success, as they sometimes favor some of the aspects which can be taken into account in the evaluation process. A similar approach could be taken for symbol recognition. However, because of the large number of variables concerning symbol recognition, it seems difficult to define a single performance measure, a suitable metric and a complete set of evaluation tests, taking into account all possibilities described in previous sections. In fact, symbol recognition remains an active research domain, and it seems more interesting to focus on the understanding of the strengths and the weakness of the existing methods rather than to attempt to measure a global performance. So it is *a priori* more suitable to consider the evaluation as a set of measures, each one corresponding to one specific aspect of recognition, determined from the characteristics of symbol recognition methods and data presented in sections 2 and 3. In a first time, it makes sense to measure each aspect alone, as some recognition methods are sometimes very specific and thus not able to deal with all of them. In a second time, some of these stand-alone criteria could be combined in order to get more global measures.

Then, the problem is which measures provide a good evaluation of symbol recognition. It can be considered that the information we expect in all cases is the label of the symbols represented in a test image. In a segmented image, that may be enough, but for a non-segmented one, including possibly several instances of different symbols, this can not be satisfactory. In this case, labels of symbols have to come with at least information as location, orientation and scale. Then, some metrics should also be defined in order to manage the accuracy of these measures. From a general viewpoint, other metrics can be taken into account to test other evaluation aspects:

- The number or rate of false positives and missing symbols.
- Considering second/third candidates and confidence rate, if available.
- Computation time: we can consider only recognition time, or we can also include the time used by other processes, such as learning, vectorization or segmentation.
- Scalability, *i.e.* how the performance degrades as the number of symbols increases. We can measure it according to the degradation of recognition rates or according to the computation time.

5 The Framework for Evaluation

Whatever the evaluation criteria and data, an evaluation framework must provide formats and tools allowing to exchange information about models, images, tests and results. It must also define a protocol to test a given method on the dataset. The first issue is about file formats of images. One assumption to be made is that image formats must not degrade the original image and must be freely available for all participants. We must work with two main kinds of formats: bitmap and vectorial. The bitmap format is already associated to a lot of solutions (such as TIFF, BMP or PNG format), even if some of these formats are more popular than other ones. For the vectorial side, some "standard" ways of representation also exist, such as DXF or more recently SVG. But for the evaluation purpose, it seems that these formats are maybe too sophisticated. In fact, a simple format, as the VEC format proposed by Chhabra [4] seems to be sufficient. Moreover, its simplicity allows its eventual extension, if required.

The evaluation framework must also define several other file formats in order to describe precisely the data included in a test and the results achieved by a method. According to the kind of evaluation, several solutions may be suitable. Indeed, the goal is to find the best compromise allowing to express all useful information without obliging the participants in interfacing their recognition methods with too sophisticated formats. The easier is probably to use some simple text files with a syntax close to those proposed by the `.ini` files, as it does not require any sophisticated parsing. However, when the information to describe becomes more complex (for example, the description of results with several measures for each image) XML seems to be an interesting flexible format. Moreover, the use of a DTD or a scheme helps to normalize data, avoiding most of description troubles. And associated with the XSLT style-sheets, it allows the extraction and the filtering of data, that can be automatically processed, both for participants (for interfacing purpose) and organizers (for automatic analysis).

Another basic idea is to give each participant the possibility to choose which tests he wants to compete in, according to the features of his method. To achieve this, each test has to be considered as a stand-alone part. Therefore, it has to contain all the information that a participant need to know about a test: which are the models involved in the test and which are the test images. This principle is also useful in some other situations. Thus, if a program crashes during a test, it is able to run the other tests.

Finally, we want to point out that the availability of the framework (formats, data, etc.) is very important. In the context of the organization of a contest, information about formats and data is required for preparing the methods and learning purposes. But from a more general point of view, a contest is nothing else than a special occasion to evaluate performances. As noticed in [18], there is a need for a corpus and performance evaluation methods, in order to evaluate symbol recognition capabilities. Since GREC'97, this need still exists and there is still a lack of public domain data with associated ground-truth.

6 The Contest on Symbol Recognition at GREC'03

The contest on symbol recognition organized in the context of GREC'03 [11] has to be seen as a first step trying to give an answer to some of the questions exposed in the previous sections. It aimed to set up the basis for the definition of a general framework for performance evaluation of symbol recognition methods. According to the general considerations stated before, the main goal of the contest was not to give a single performance measure for each method, but to provide a tool to compare different symbol recognition methods under several criteria.

As we wanted to promote a wide participation of the graphics recognition research community and because of the large number of possible options to be considered for defining the data set and the protocol of evaluation, a questionnaire was designed to get feedback from the potential participants. The analysis of the answers to this questionnaire helped us to take the final choices.

The first important decision was to use synthetic data for the contest. We decided to automatically generate all images because this way, we were able to generate a large amount of data and its ground-truth. Concerning the kind of data, we decided to focus on symbols composed only of linear graphic primitives (straight lines and arcs). We did not consider symbols containing solid shapes or textures. The reason was to simplify, in this first edition of the contest, the description of data, and to be able to provide vectorial data for all symbols. We only chose two application domains, architecture and electronics, which were the most used by the participants, but we mixed symbols from both domains in the datasets. We organized the data in three different sets of symbols: the first one containing only 5 symbols, the second one with 20 symbols and the third one with 50 symbols. This way we could evaluate how the performance of the methods evolved with an increasing number of models to consider. In figure 2 we can see some of the symbols used in the contest. As we decided to use synthetic data, the data set was limited to pre-segmented images of the symbols, and no real drawings were considered in this first edition of the contest. We only focused on the evaluation of recognition and not segmentation plus recognition.

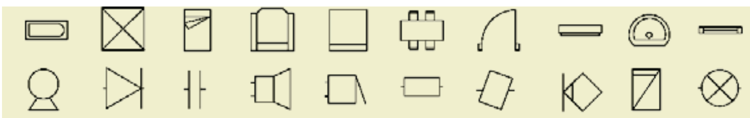


Fig. 2. A sample of 20 symbols used in the contest.

In order to favor the participation in the contest, we provided all images in both possible formats: bitmap and vectorial. Vectorial images were provided in .VEC format [8] as this is the simplest format able to represent vectorial data and is well-known by the graphics recognition community. Binary images were generated from the vector representation. There is not any vectorization process applied to binary images to get the vector representation so that results concern only recognition and not vectorization. This way, we do not introduce any alteration to data as a result of vectorization. Nevertheless, a participant could take the binary images and apply its own vectorization method.

We defined three categories for the generation of degraded images: the first one contained rotated and scaled images of each symbol. The second one aimed to model binary degradation of images such as that produced by printing, scanning or photocopying. A model based on the method proposed by Kanungo *et al.* [17] was used to generate nine different models of degradation, illustrated in figure 3. This model was applied to binary images and thus, no vector representation is available for this kind of images. Finally, the third category included images with shape distortions, similar to those produced by hand-drawing. The model for generating such images is based on the *Active Shape Models* [19], and some sample images can be seen in figure 4. This model is applied to vectorial images and thus, both vectorial and binary images are available. With these three kinds of degradation, we cover a wide range of noise which can be found in real images.

72 different tests were designed to evaluate the performance in all these cases. These tests were grouped into five categories according to the type of the data: tests containing only ideal images of the symbols, test with rotated and scaled images, tests with binary degradations, tests with shape distortions and tests combining binary degradations and shape distortions. For each category, several

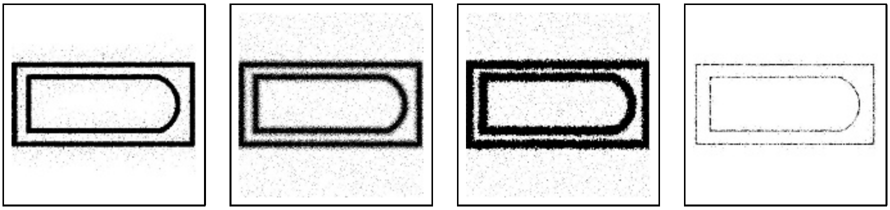
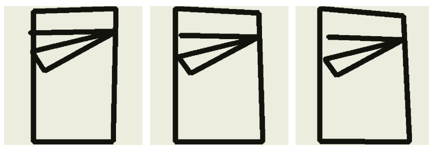


Fig. 3. Samples of some degraded images generated using the Kanungo method for some models of degradation used.



(a) (b) (c)

Fig. 4. Examples of increasing levels of vectorial distortion.

tests were designed, each one with a different number of symbols or different levels of degradation and distortion. With this organization of the tests, we could evaluate the robustness of methods to the most common types of degradations and to the increase in the number of symbols to be recognized. Before the contest, some sample tests for each category were supplied to the participants so that they could train their methods with them. Each test was independent of all the others so that each participant could choose the tests he wanted to take part in. The description of the tests was formatted as a `.ini` file because only information about the models and the images involved in the test was required. However, each participant had to provide the final results for each test in an XML file, as several parameters (label of the symbol, computation time, rotation, scale, etc.) could be provided for each image in the result file. Although the XML format of the file can include more information about the result of the recognition, in this first edition of the contest we only considered the label of the recognized symbol and the computation time.

Five methods took part in the contest. A more detailed analysis of the contest and the results achieved by each method can be found in [11].

7 Conclusion

The performance evaluation of symbol recognition is not a trivial task. Many characteristics and particularities have to be taken into account: the application domain, the graphical entities composing the symbol, the format of the data, segmentation, vectorization, noise and degradation, the definition of performance measures, etc. In this context, many precautions have to be taken. Indeed, the purpose must be to determine the strengths and weakness of symbol recognition methods. Thus, a generic and independent framework have to be proposed in order to measure, as well as possible, all these criteria. A lot of open questions arise when trying to define it, concerning the kind of data and how to generate it, the performance metrics, and the protocol of evaluation. We have synthesized some of them in this paper, those which led us to the organization of a contest on symbol recognition at GREC'03. We have also described the options taken to set up the framework for the contest, among that set of possibilities.

This framework should be considered only as a starting point in the definition of a general framework for the evaluation of symbol recognition. We are aware that in this first edition of the contest we had to choose simple options and we left out many interesting issues. Thus, there are several points which should be considered in order to make this framework more general and complete:

– Collecting more images from different application domains, in order to get a really large dataset, with symbols including all kinds of graphic primitives, and thus to be able to measure the scalability of the methods. We should organize this dataset according to the application domain, the shape of the symbols or some other criteria.
– The relation of recognition with other processes, such as vectorization or segmentation, should be clearly determined in order to establish how these processes influence the performance of recognition. Particularly, we should provide non-segmented images and define how to evaluate them.
– More models of image degradation and distortion can be studied and used to generate noisy images.
– Performance measures, other than the label of the symbol, should be considered, such as orientation, scale, ability to segment, computation time, etc. How to combine all these measures also remains as an open question.
– Finally, the evaluation framework should also provide tools for the automatic evaluation of any recognition method, and as much as possible be permanently available in order to constitute a tool of reference. We are planning to provide this service thanks to a forthcoming web site.

References

1. Haralick, R.: Performance characterization in image analysis: Tninning, a case in point. Pattern Recognition Letters **13** (1992) 5–12
2. Guyon, I., Haralick, R., Hull, J., Phipliops, I.: Data sets for ocr and document image understanding research. In Bunke, H., Wang, P., eds.: Handbook of Character Recognition and Document Image Analysis. World Scientific (1997) 779–800
3. Kong, B., Phillips, I.T., Haralick, R.M., Prasad, A., Kasturi, R.: A benchmark: Performance evaluation of dashed-line detection algorithms. In Kasturi, R., Tombre, K., eds.: Graphics Recognition: Methods and Applications, Selected Papers from First International Workshop on Graphics Recognition, GREC'95. Springer, Berlin (1996) 270–285 Volume 1072 of Lecture Notes in Computer Science.
4. Chhabra, A., Phillips, I.: The second international graphics recognition contest - raster to vector conversion: A report. [20] 390–410

5. Chhabra, A., Philips, I.: Performance evaluation of line drawing recognition systems. In: Proceedings of 15th. International Conference on Pattern Recognition. Volume 4. (2000) 864–869 Barcelona, Spain.
6. Wenyin, L., Zhai, J., Dori, D.: Extended summary of the arc segmentation contest. In Blostein, D., Kwon, Y., eds.: Graphics Recognition: Algorithms and Applications, Selected Papers from Fourth International Workshop on Graphics Recognition, GREC'01. Springer, Berlin (2002) 343–349 Volume 2390 of Lecture Notes in Computer Science.
7. Wenyin, L., Dori, D.: A protocol for performance evaluation of line detection algorithms. Machine Vision and Applications **9** (1997) 240–250
8. Phillips, I., Chhabra, A.: Empirical performance evaluation of graphics recognition systems. IEEE Transactions on Pattern Analyisis and Machine Intelligence **21** (1999) 849–870
9. Hilaire, X.: A matching scheme to enhance performance evaluation of raster-to-vector conversion algorithms. In: Proceedings of 7th International Conference on Document Analysis and Recognition. Volume 1. (2003) 629–633 Edinburgh, Scotland.
10. Aksoy, S., Ye, M., Schauf, M., Song, M., Wang, Y., Haralick, R., Parker, J., Pivovarov, J., Royko, D., Sun, C., Farneboock, G.: Algorithm performance contest. In: Proceedings of 15th. International Conference on Pattern Recognition. Volume 4. (2000) 870–876 Barcelona, Spain.
11. Valveny, E., Dosch, P.: Symbol recognition contest: A synthesis. In: Selected Papers from Fith International Workshop on Graphics Recognition, GREC'03. (2004) To appear.
12. Chhabra, A.K.: Graphic Symbol Recognition: An Overview. [20] 68–79
13. Cordella, L.P., Vento, M.: Symbol recognition in documents: a collection of techniques? International Journal on Document Analysis and Recognition **3** (2000) 73–88
14. Lladós, J., Valveny, E., Sánchez, G., Martí, E.: Symbol Recognition: Current Advances and Perspectives. In Blostein, D., Kwon, Y.B., eds.: Graphics Recognition – Algorithms and Applications. Volume 2390 of Lecture Notes in Computer Science. Springer-Verlag (2002) 104–127
15. Tombre, K., Ah-Soon, C., Dosch, P., Masini, G., Tabbone, S.: Stable and Robust Vectorization: How to Make the Right Choices. In Chhabra, A.K., Dori, D., eds.: Graphics Recognition—Recent Advances. Volume 1941 of Lecture Notes in Computer Science. Springer-Verlag (2000) 3–18
16. Baird, H.S.: The State of the Art of Document Image Degradation Modeling. In: Proceedings of 4th IAPR International Workshop on Document Analysis Systems, Rio de Janeiro (Brazil). (2000)
17. Kanungo, T., Haralick, R.M., Baird, H.S., Stuetzle, W., Madigan, D.: Document Degradation Models: Parameter Estimation and Model Validation. In: Proceedings of IAPR Workshop on Machine Vision Applications, Kawasaki (Japan). (1994) 552–557
18. Tombre, K., Chhabra, A.K.: General Conclusions. [20] 411–420
19. Cootes, T., Taylor, C., Cooper, D., Graham, J.: Active shape models: Their training and application. Computer Vision and Image Understanding **61** (1995) 38–59
20. Tombre, K., Chhabra, A.K., eds.: Graphics Recognition—Algorithms and Systems. Volume 1389 of Lecture Notes in Computer Science. Springer-Verlag (1998)