HAL
archives-ouvertes.fr

ARG based on arcs and segments to improve the symbol recognition by genetic algorithm

Jean-Pierre Salmon, Laurent Wendling

▶ To cite this version:

HAL Id: inria-00184880
https://hal.inria.fr/inria-00184880

Submitted on 2 Nov 2007

ARG based on arcs and segments to improve the symbol recognition by genetic algorithm

J.-P. Salmon and L. Wendling

LORIA-UMR 7503, INPL,
BP 239, 54506 Vandoeuvre-lès-Nancy, France
`{salmon,wendling}@loria.fr`

Abstract. In this paper a genetic matching scheme is extended to take into account primitive arcs and complex description in the pattern recognition process. Classical ways only focus on segments and are sensitive to over segmentation effect. Our approach allows to improve the recognition by handling more accurate description and also to decrease processing time by limiting the number of vertices to match. Experimental studies using real data attest the robustness of our approach.

Keywords: graphic recognition, vector representation, inexact graph matching, technical drawings.

1 Introduction

Symbol recognition is especially well suited to using structural pattern recognition techniques, unlike usual statistical classification techniques. When dealing with specific families of symbols, techniques similar to OCR could be used; this is the case for symbols having all a loop [1] or for music recognition. However these techniques have their own limitations, in terms of computational complexity and of discrimination power, and in the present case we have a family of symbols which is much too large and diverse for these techniques to be suitable. Actually, graph matching techniques are especially adapted to the specificities of symbol recognition[2]. Generally, methods match symbols to be recognized with model graphs using graph distance computations. In its basic principle, it is sensitive to errors and noise; as we usually cannot assume that segmentation is perfect nor reliable, this means that the graphs to be processed can also have a number of extra or missing vertices. To deal with this problem, error-tolerant subgraph isomorphism algorithms have been proposed [3, 4]. Another problem with graph matching is the computational complexity of subgraph isomorphism methods. To deal with this, even if a lot of efforts have therefore been devoted to optimizing the matching process through continuous optimization [5] or constraint propagation techniques to perform discrete [6, 7] or probabilistic relaxation [8].
Arcs and circles are basic elements contained in technical drawings (engineering, electrical networks, mechanical...) and often relate to parts of symbols (for

instance diode or logical nor representations) [9]. Considering raster data, powerful operator dedicated to their recognition should be required for both analysis and understanding of documents [10–12]. In the early years, many approaches have been developed (or extended) to extract such primitives [13, 14] as arc-fitting methods [15, 16], using Hough transform [17, 18] and stepwise arc extension methods [19, 20, 13]. We can also denote that several methods were embedded in comparative studies using GREC contest [21, 22]. For instance the last edition has shown than RANVEC method [23] supersedes most of them in many cases. This approach has been used here to ensure an accurate description from raster image.

In this paper we propose to integrate primitive arcs and segments in Attributed Relational Graph. Second step consists in matching such ARGS with a genetic algorithm to improve the recognition while keeping a faster processing.

2 Graph Description

2.1 ARG Structure

The vertices of our Attributed Relational Graph represent the graphic pritimives and the edges characterize the topological relations between them. First, we consider the contour representation provides in [24] to handle with filled symbols (see figure Fig. 1).
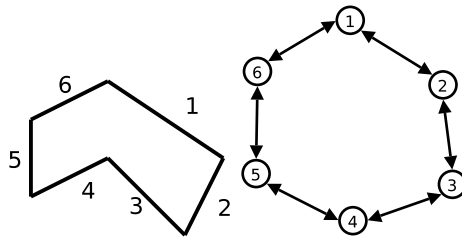

Fig. 1. A contour vector and its graph.

In the figure Fig. 1 the associated planar graph is entirely connected, that is each vertex is linked to two others. The variability of symbols encountered in graphic documents, requires to extend such process to take into account symbols split into strings of points. Obviously vertex constraints are weaker: a vertex can be connected at least to one other (see 2).

This kind of representation does not allow to really take into account symbols composed of several connected components (see fig. Fig. 3).

The noise presents can remove some jonctions in image corresponding to remove some edges in the ARG. To fix this problem, lookup areas has been defined around ends of graphical primitives to locate the nearest graphical pritimives.
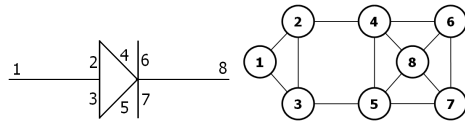
Fig. 2. A Symbol composed of several strings and its associated graph.

In this way, if a segment is isolated from the others then a particular relation is created to link it. The lookup areas is spread-out in order to connected all the graphical primitives. Thus, all the vertices of the ARG are connected at less with an other.
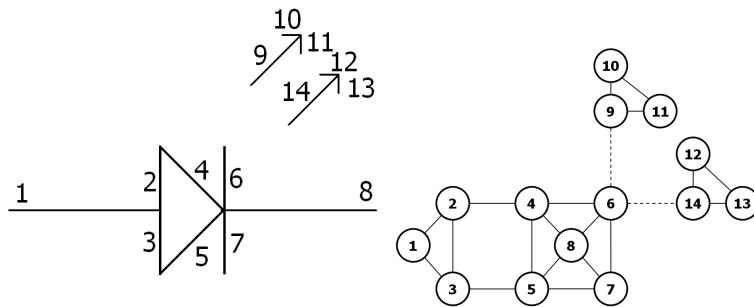

Fig. 3. A symbol composed of 3 connected components and its associated graph.

This graph provides a more accurate description of symbols. Nevertheless symbols may have arcs in their composition. Generally they are approximated by a set of segments depending on the scale (Fig. 4).
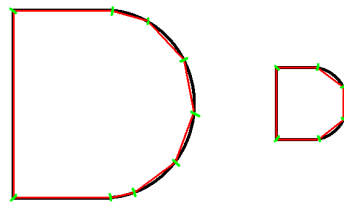

Fig. 4. Problem of vectorization : increase in the number of segments due to scale factor.

The ability to handle with arcs has been studied to improve the accuracy during the recognition step.

2.2 Attributes of ARG (features of arcs and segments)

A set of sharing features is required in order to be able to compare arcs and segments when vectorization errors remain. Unary features are usually process within the recognition step to characterize graphic primitives, as the length, the sum of internal angles in relation to adjacent primitives and the distance between the middle of primitive and the middle of the associated chord (equal to 0 in case of segment) (see Fig. 5).
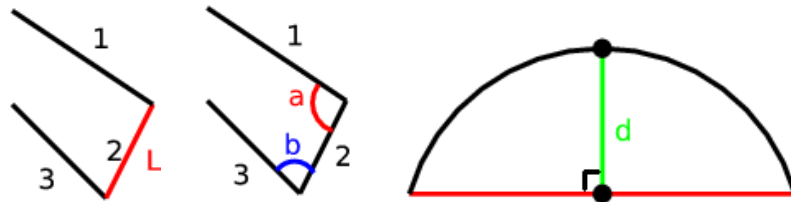

Fig. 5. Common vertex features: length, sum of internal angles, distance between the primtives and its chord.

Binary relations between primitives are also used as angle between two primitives, distance between middle of primitives, angles between farer points of primitives (see Fig. 6).
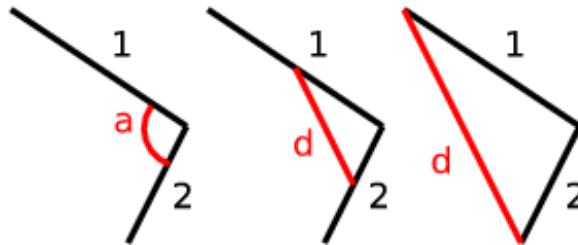

Fig. 6. Edge features: the angle, distance between center points, the maximum distance between end points

Determining an angle between a segment and an arc has no sense. So we consider the angle between the segment and an adjoining segment of the arc (see Fig. 7).

All these features are rotation and translation invariants. The acute angle between two primitives is invariant to scale factor as well as the sum of internal angles. Li [25] proposes to integrate a logarithm function to ensure scale factor invariance of sizes and distances. Other ways focus on normalized height and distance following maximal values computing on the shape under consideration.
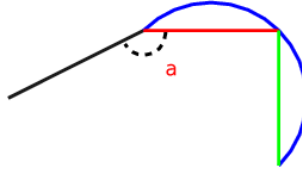
Fig. 7. Angle between a segment and a adjoining segment of an arc.

Nevertheless these approaches are sensitive to segmentation errors. Here the fitness function, proposed Khoo and Suganthan [24], is adapted to efficiently take into account the scale factor. Such description allows to efficiently differentiate the shapes provided figure Fig. 8 while preserving a suitable recognition despite unexpected vectorization errors.
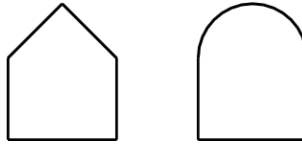

Fig. 8. Similar description with ambiguities between an arc and two unexpected segments.

3   Genetic Algorithm

In previous part, we presented a description composed of segments and arcs, which is more robust and accurate to describe complex symbols, especially for scale sensitive symbols. Such description is structured into an ARG.

Nevertheless considering real data, noises due to the scan can disturb original signal while adding unexpected artifacts and defomations into the ARG representation. Therefore, the recognition method should be able to match the sub-graphs and to be error-tolerant. That the reason why, we chose the genetic algorithm.

3.1   Fitness function

For any feature of the vertex, the matching cost is calculated as follows:

$$C_{i,f_i,u} = |a_{f_i,u} - a_{i,u}| \tag{1}$$

with f_i the vertex matched with the vertex i following the feature u. Let us consider a vertex i, the global cost A_{i,f_i} is calculated as follows:

$$A_{i,f_i} = \begin{cases} \sum_{u=1}^{U} \tanh(k_u(t_u - C_{i,f_i,u})), & f_i \neq Null \\ 0 & Otherwise \end{cases} \qquad (2)$$

with U the number of features. This sum relies on the global matching cost obtained from vertex i. The use of function $tanh$ allows to normalize values between -1 and 1 and to act on the importance of each feature. The parameter k relies on the slope of $tanh$ and t is a threshold assuming the matching cost is too high to consider symbols are similar. We introduce a close process to handle with edges (see eq. 3 and 4 next page) with i, j adjacent vertices and b an edge feature.

$$C_{i,j,f_i,f_j,b} = |a_{f_i,f_j,b} - a_{i,j,b}| \qquad (3)$$

$$D_{i,j,f_i,f_j} = \begin{cases} \sum_{b=1}^{B} \tanh(k_b(t_b - C_{i,j,f_i,f_j,b})), & f_i \neq Null, f_j \neq Null, \\ 0 & Otherwise \end{cases} \qquad (4)$$

The fitness function became:

$$F = \sum_{i=1}^{S} A_{i,f_i} + \lambda \sum_{i=1}^{S} \sum_{j=i+1}^{S} D_{i,j,f_i,f_j} \qquad (5)$$

with S the number of the vertices. A value of fitness greater than 0 implies vertices are matched, equal to zero relies on unknown information (NULL) and lower than 0 implies a bad match. Parameter λ weights the importance of vertex features with respect to edge features during the initialization step.

3.2 Matching

In our application a gene is associated to the matching between the vertex of a scene and those of the model. Strings of genes are randomly set in the initialization step. More the scene is complex (following the number of connected components and the number of primitives)and more the string is long. Following the generation, crossovers are carried out with the strings and new genes are randomly added. The crossovers consist in randomly swapping genes between two strings. When the number of generations is completed, only genes having the best quality according to the fitness function are kept. A number of generations is preset to guarantee an acceptable response time. A way to provide optimal values of the algorithm parameters is proposed in experimental section § 4.

We have studied two approaches of matching. First, we compared a scene to each model, one by one, using the genetic algorithm. Also, for each model, a similarity value and the best matching (vertex to vertex),is obtained. The algorithm returns the model having the greatest similarity value.

In a second way, one scene is compared to all the models in a unique step and the most similar parts of models are kept. Also, for each vertex of the scene, the vertex of the model having the greatest similarity value is selected. Such method is interesting to detect parts (or symbols) included in the composition of other symbols (see figure Fig. 9). Obviously this method relies on the number of symbols contained in the database and is sensitive to scale. Nonetheless the method is able to process these two cases.
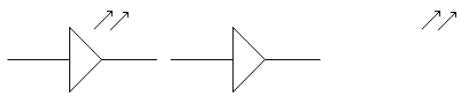
Fig. 9. a) scene composed from 2 models b) first model (5 matched vertices) c) second model (6 matched vertices).

4   Experimental study

4.1   Parameters k and t setup

Let us consider a set of learning samples.The study leaded to find optimal values for parameters k and t has been performed as follows: Let us consider p discrete values for k and n queries. We keep the value of k which maximizes the percentage of right matching. Then the value of t is determined in a same way from k. For example the figure Fig. 10 shows the recognition rates obtained for k and t using $n = 1000$ queries. Optimal values should be set a 0.1 for k and 5 for t.

4.2   Variability of the results

The accuracy of the method relies on the number of generations. The influence of such parameter was studied to improve the robustness of our approach by refining the stopping criterion of the basic algorithm. For instance let us consider n queries (for example $n = 1000$ see figure Fig. 11.a), we can remark a minimal of generations allowing to achieve a maximal recognition rate (here 300).

It is well know that when one uses stochastic optimization algorithms results obtained in a single experiment do not guarantee repeatability and are not significant. That the reason why, we have also determine how many queries are
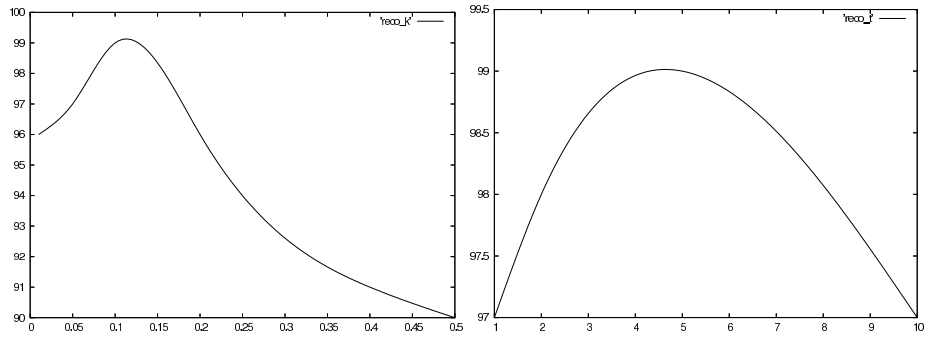
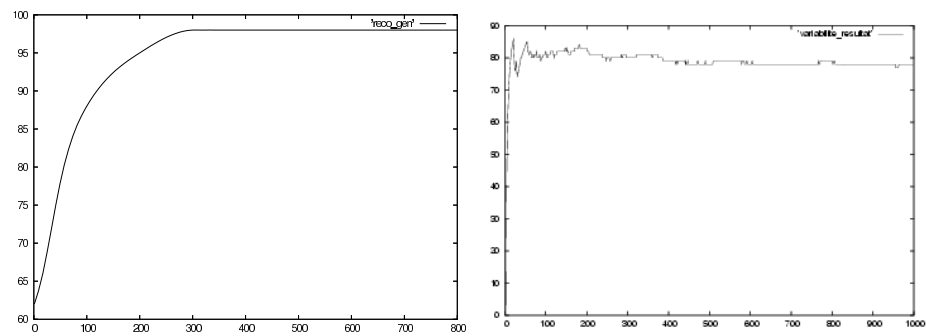**Fig. 10.** Recognition rates following t and k.

Fig. 11. Recognition rates following the number of generation (left) and cumulated recognition rates (right).

enough to keep a stable recognition rate by calculated a cumulated mean as follows:

$$rate(n) = \frac{((n-1) * rate(n-1) + \delta(n))}{n}$$

with: $rate(n)$ the mean recognition rate after n queries and $\delta(n)$ is equal to 1 if the query n is successful,0 otherwise. Figure Fig. 11.b shows that recognition rates are relatively stable about 100 queries.

4.3 Experimental study

In this sub-section, all recognition rates are computed with the results of previous studies.

First we have tested our approach with symbols made of several strings of points using GREC database (distortion 2). We have considered 7 classes merged with 8 other classes in the dataset of 74 symbols (following the class, the number of images is different) to show the robustness of the approach. Figure Fig. 12 presents target classes and recognition rates A mean rate around 93.2% is reached. Such results underline the interest of our decomposition scheme.
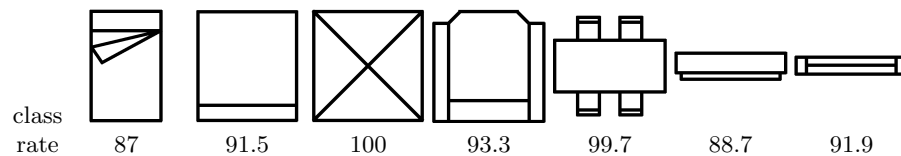

class							
rate	87	91.5	100	93.3	99.7	88.7	91.9

Fig. 12. Recognition rates.

To show the interest of handling with primitive arc, another test has been performed using a database composed of 9 classes of 11 symbols by class having arcs. A random model of deformation has been applied to the extremities of the primitives (see figure Fig. 13).
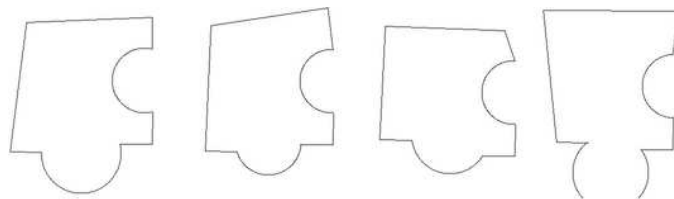

Fig. 13. Examples of distorted symbols.

The basic approach [24] gives rise to a mean recognition rate of 52.1% (with classical vectorization in a set of segments). Handling with arcs in the calculation of fitness allows to reach a mean recognition rate of 90.4% (see figure Fig. 14).
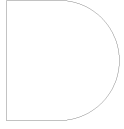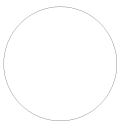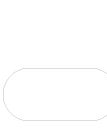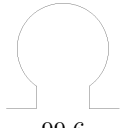
class					
before	78.4	47.6	9.5	7.3	30.2
after	83.6	95.8	79.3	100	72.1

class				
before	31.1	99.6	77.3	87.9
after	89.2	98.9	99.7	94.7

Fig. 14. Recognition rates obtained.

5 Conclusion and future work

In this paper, we introduced an Attributed Relational Graph able to describe the symbols composed of several strings of points and several connected components. Likewise, we used the segments and the arcs to describe more accurately the symbols and also to avoid the mistakes of matching. Common invariant features to geometric transformations have been proposed to take into account both primitives in the ARG. We demonstrated than a genetic algorithm based on this ARG provides better results. In our future work, we want to take into account more complexe graphical primtives.

References

1. Okazaki, A., Kondo, T., Mori, K., Tsunekawa, S., Kawamoto, E.: An Automatic Circuit Diagram Reader with Loop-Structure-Based Symbol Recognition. IEEE Transactions on PAMI **10**(3) (1988) 331–341
2. 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
3. Bunke, H.: Error Correcting Graph Matching: On the Influence of the Underlying Cost Function. IEEE Transactions on PAMI **21**(9) (1999) 917–922
4. Messmer, B.T., Bunke, H.: A New Algorithm for Error-Tolerant Subgraph Isomorphism Detection. IEEE Transactions on PAMI **20**(5) (1998) 493–504
5. Kuner, P.: Efficient Techniques to Solve the Subgraph Isomorphism Problem for Pattern Recognition in Line Images. In: Proceedings of 4th Scandinavian Conference on Image Analysis, Trondheim (Norway). (1985) 333–340
6. Habacha, A.H.: Structural Recognition of Disturbed Symbols Using Discrete Relaxation. In: Proceedings of 1st International Conference on Document Analysis and Recognition (Saint-Malo, France). Volume 1. (1991) 170–178
7. Wilson, R.C., Hancock, E.R.: Structural Matching by Discrete Relaxation. IEEE Transactions on PAMI **19**(6) (1997) 634–648

8. Christmas, W.J., Kittler, J., Petrou, M.: Structural Matching in Computer Vision Using Probabilistic Relaxation. IEEE Transactions on PAMI **17**(8) (1995) 749–764
9. Kasturi, R., Bow, S., ELMasri, W., Shah, J., Gattiker, J., Mokate, U.: A system for interpretation of line drawings. IEEE Trans. on Pattern Analysis and Machine Intelligence **12(10)** (1990)
10. Cordella, L.P., Vento, M.: Symbol recognition in documents: a collection of techniques? International Journal on Document Analysis and Recognition **3**(2) (2000) 73–88
11. Messmer, B.T., Bunke, H.: Automatic learning and recognition of graphical symbols in engineering drawings. Graphics Recognition: Methods and Applications (1996) 123–134
12. Tombre, K., Dori, D.: Interpretation of engineering drawings. Handbook of Character Recognition and Document Image Analysis (1997) 457–484
13. Song, S., Lyu, M.R., Cai, S.: Effective multiresolution arc segmentation: Algorithms and performance evaluation. IEEE Trans. on Pattern Analysis and Machine Intelligence **16(11)** (2004) 1491–1506
14. Wenyin, L., Zhai, J., Dori, D.: Extended summary of the arc segmentation contest. In Blostein, D., Kwon, Y.B., eds.: Postproceedings of GREC'02. Volume 2390 of LNCS., Springer-Verlag (2002) 343–349
15. Rosin, P., West, G.: Segmentation of edges into lines and arcs. Image and Vision Computing **7(2)** (1989) 109–114
16. Elliman, D.: An algorithm for arc segmentation in engineering drawings. In Blostein, D., Kwon, Y.B., eds.: Graphics Recognition Algorithms and Applications. Volume 2390 of LNCS., Springer-Verlag (2002) 350–358
17. Conker, R.: A dual plane variation of the hough transform for detecting nonconcentric circles of different radius. Computer Vision and Image Processing **43** (1988) 115–132
18. Leavers, V.: The dynamic generalized hough transform: Its relationship to the probabilistic hough transforms and an application to the concurrent detection of circles and ellipses. Computer Vision, Graphics, Image Understanding **56(3)** (1992) 381–398
19. Dori, D.: Vector-based arc segmentation in the machine drawing understanding system environment. IEEE Trans. Pattern Analysis and Machine Intelligence **17(11)** (1995) 1057–1068
20. Liu, W., Dori, D.: Incremental arc segmentation algorithm and its evaluation. IEEE Trans. Pattern Analysis and Machine Intelligence **20(4)** (1998) 424–431
21. http://www.cs.cityu.edu.hk/liuwy/ArcContest/Arc SegContest.htm: GREC'01 contest. Proceedings of 4th IAPR International Workshop on Graphics Recognition, Kingston, Ontario (Canada) (2002)
22. http://www.cvc.uab.hk.es/grec03/contest.htm: GREC'03 contest. Proceedings of 5th IAPR International Workshop on Graphics Recognition, Barcelona (Spain) (2004)
23. Hilaire, X.: Ranvec and the arc segmentation contest: Second presentation. In Liu, W., Llados, J., eds.: Postproceedings of GREC'05. LNCS, Springer-Verlag, to appear (2006)
24. Khoo, K., Suganthan, P.: Evaluation of genetic operators and solution representations for shape recognition by genetic algorithm. Pattern Recognition Letters **23**(13) (2002) 1589–1597
25. Li, S.Z.: Matching: Invariant to Translations, Rotations and Scale Changes. Pattern Recognition **25**(6) (1992) 583–594