Learning of Structural Descriptions of Graphic Symbols using Deformable Template Matching

Ernest Valveny, Enric Martí
Computer Vision Center - Computer Science Department
Universitat Autònoma de Barcelona
Edifici O - Campus UAB. 08193 Bellaterra. Spain
{ernest|enric}@cvc.uab.es

Abstract

Accurate symbol recognition in graphic documents needs an accurate representation of the symbols to be recognized. If structural approaches are used for recognition, symbols have to be described in terms of their shape, using structural relationships among extracted features. Unlike statistical pattern recognition, in structutal methods, symbols are usually manually defined from expertise knowledge, and not automatically infered from sample images. In this work we explain one approach to learn from examples a representative structural description of a symbol, thus providing better information about shape variability. The description of a symbol is based on a probabilistic model. It consists of a set of lines described by the mean and the variance of line parameters, respectively providing information about the model of the symbol, and its shape variability. The representation of each image in the sample set as a set of lines is achieved using deformable template matching.

1. Introduction

Structural recognition of distorted patterns requires an structural representation of the pattern to be recognized. This representation must be able to code information about all possible shapes of the pattern, including distortions and changes due to multiple factors: intrinsic variability of the pattern, noisy images, errors introduced in feature extraction, etc. Moreover, many times the frontiers between valid and invalid shapes of a pattern are not always clear. Then, pattern representation must also face uncertainty of the ambiguous shapes.

In graphics recognition, structural pattern recognition - usually based on graph matching - have been widely used to identify distorted symbols in graphic documents [1, 5, 6, 7]. In these methods, information about the shape of the symbol is coded with the description of an ideal model of the symbol, consisting of structural relationships among a set of structural primitives or features, such as lines, points, junctions, etc. Different shapes of the symbol are generated by applying a set of rules modifying the representation of the model. Each rule is associated with an application cost and thus, any generated distorted shape can also be associated with a global distortion cost. Depending on this cost, the shape is taken as a valid representation of the symbol or not.

In this way, a symbol is represented by three factors: the description of the ideal model, the set of generation rules and the cost associated with each rule. The set of generation rules is fixed for all symbols, but the ideal model and the cost function are specific to each symbol and thus, they could be learned from a set of sample images for each symbol.

Learning from a set of examples has been widely developed in many areas of pattern recognition. In statistical pattern recognition it is often an intrinsic process in the design of a classifier. But research in structural pattern recognition have focussed more on recognition than on learning the representation of the pattern, due to the difficulty of generalizing structural descriptions from a set of examples. Specifically in graphic symbol recognition, both the model of the symbol and the cost function are usually manually predefined as a previous step to recognition, using expertise knowledge. Recently, some efforts have been carried out to develop methods for getting more optimal model representations using automatic learning techinques.

In this way, Jiang et al. [4] have investigated the concept of generalized median as a representative of a class of symbols. The generalized median is defined as the pattern $\bar{p}$, belonging to the set of all possible patterns U, which minimizes the sum of distances to all patterns from a sample set S:

$$\bar{p} = \arg\min_{p \in U} \sum_{q \in S} d(p, q) \qquad (1)$$

They apply this concept to a graph representation, using a genetic algorithm to find the generalized median graph from a set of samples of graphical symbols of electronic diagrams. In this way, the model of the pattern is learned, but the cost function associated with edit operations in graph matching is still manually predefined.

Another approach to learning structural descriptions has been proposed by Cordella et al.[2]. They describe the graph representation of a symbol with a set of logic predicates. Then, they apply Inductive Logic Programming to modify these predicates from the set of examples, which are also described with a set of predicates.

In this work we present an approach to learning structural descriptions of symbols, which allows to get both a representation of the model of the symbol, and an estimation of the parameters used to define the cost function. The description of the symbols is based on a probabilistic framework, in which any distorted shape generated from the model of the symbol is given a probability of being a valid shape of that symbol. This probability is used to compute the cost function. Then, learning from the set of samples the parameters of the probabilistic model provides both the representation of the model and the parameters needed for computing the cost function.

This approach is based on deformable template matching for matching an image with the model of the symbol. Its application to symbol recognition has been reported in [8]. Symbols are described by a set of lines, not necessarily connected. Each line is defined using three parameters: mid point coordinates, orientation and length. Learning is carried out by adjusting, using deformable template matching, each sample image with an instance of this set of lines. Then, mean and variance for each of the parameters of the lines can be calculated. The mean of each line is used to get the representation of the model for the symbol, while the variance can be used to define the cost associated to each deformation.

The paper is structured as follows. In section 2 we show how deformable template matching is used to match an unknown image with the model of a symbol, getting a lineal description of the image. Then, section 3 is devoted to explain how the parameters used to represent each symbol are learned from the lineal descriptions of all sample images. Finally, some results are shown and discussed in section 4 while final conclusions are reported in section 5.

2. Structural description of an image using deformable template matching

The first step in our learning approach consists in getting the representation of every sample image. We use deformable template matching to get this representation. In deformable template matching [3], an initial representation

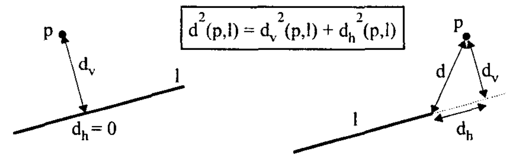

Figure 1. Distance between a point and a line.

of a pattern is deformed in order to fit the input image. Deformation is guided by the joint action of two opposite forces: external force, trying to deform the model to the image and internal force, trying to restore the model to its initial shape. This process is modeled as the minimization of an energy function, composed of two terms: internal energy and external energy. The final result of the deformation process is the best representation of the input image as an instance of the initial pattern.

Then, we start with a predefined representation of the symbol, and we apply deformable template matching, to match every sample image with this model. As a result, we get a deformation of the model, corresponding to the best representation of the image. The description of an image consists of a set of straight lines. Each line is defined by three parameters: midpoint position, orientation and length. All images of a given symbol are described using a fixed number of lines.

Internal energy is defined assuming that each of the parameters follows a gaussian probability distribution centered on initial values of the parameters. Then, internal energy models the probability of a given deformation of the model of being a valid shape of that symbol. From that probability, and taking the negative log, we get the following expression for internal energy:

$$E_{int} = \sum_{i=1}^{n} \left(\frac{t_{x_i}^2}{2\sigma_{t_{x_i}}^2} + \frac{t_{y_i}^2}{2\sigma_{t_{y_i}}^2} + \frac{\sin^2\theta_i}{2\sigma_{\theta_i}^2} + \frac{s_i^2}{2\sigma_{s_i}^2} \right) + K \quad (2)$$

where n is the number of lines in the symbol; t_{x_i}, t_{y_i}, θ_i and s_i are the changes in parameters of midpoint position, orientation and length, respectively, for line i; $\sigma_{t_{x_i}}$, $\sigma_{t_{y_i}}$, σ_{θ_i} and σ_{s_i} are the standard deviations for midpoint position, orientation and length for line i; and K is a constant.

External energy is defined as a distance from the image to a given representation of the symbol. It is also defined using a probabilistic approach. Each line in the model is assumed to be able to generate points in the image following a gaussian distribution based on the distance of the point p to the line l. This distance is computed taken into account the distance between the point and the line and the difference in orientation between the point and the line through the following expression:

$$d(p,l) = (1 + k_\alpha \cdot \sin^2(\alpha_p - \alpha_l)) \cdot d_{pos}(p,l) \quad (3)$$

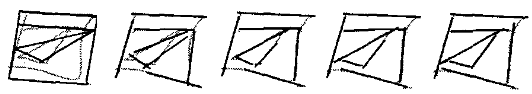

Figure 2. Deformation of the model (in black) to fit a sample image (in gray).

where $d_{pos}(p, l)$ is the distance between p and l, computed as shown in figure 1; α_l is the orientation of line l; α_p is the orientation of the line passing through point p. It is measured from the analysis of the neighborhood of p; and k_α is a factor which controls the relative weighting between orientation and position factors.

Thus, the probability that an image I has been generated by a deformed representation of a symbols, S' is:

$$P(I|S') = \prod_{i=1}^{n} \sum_{j=1}^{m} \frac{1}{2\pi\sigma} e^{-\frac{d^2(p_i,l_j)}{2\sigma^2}} \tag{4}$$

where p_i stands for each of image pixels and l_j for every line of the symbol. σ is the standard deviation of the probability of generation.

External energy is also defined as the negative log of this probability. Minimization of global energy (combination of internal and external energy) is not an straightforward problem. It can be seen as a problem with incomplete or missing information and then, an implementation of the EM algorithm can be used to solve it, reformulating it as a problem with complete information. In the expectation step, for each point p_i, and each line l_j, the probability π_{ij} of p_i of having been generated by l_j is estimated. With this estimation, external energy can be expressed in the following way:

$$E_{ext} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} \pi_{ij} \cdot d'^2(p_i, l_j) \tag{5}$$

With this expression, in the maximization step we can find the new values for the parameters of each line by direct minimization of the new energy function, finding successively, orientation, midpoint position and length. When the algorithm converges, the final result is the valid representation of the symbol with the highest probability of being able to generate the image. More details about the process of adjusting an image by a model of the symbol using deformable template matching can be found in [8]. Figure 2 shows an example of how the model of the symbol is deformed to fit the image.

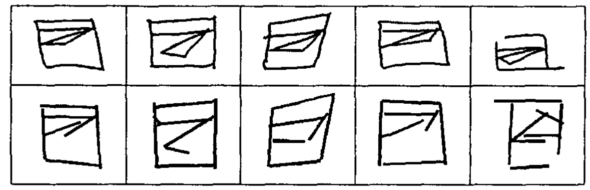

Figure 3. Sample images (above) and their bad lineal representation (below).

3. Learning of the representation of a symbol

The starting point for learning is the set of sample images and an initial representation of the model of the symbol. For this initial representation, manually predefined values are assigned to the parameters of each of the lines composing the symbol. The sample images are labeled with the symbol to which they correspond.

The first step consists in getting a lineal representation of every sample image. Thus, each sample image is skeletonized and matched with the predefined model of its corresponding symbol using the deformable template matching approach described in previous section. As a result, we get the lineal representation of the symbol corresponding to each image. In this first step we have to be careful with bad adjustments of the model to certain images, as it is shown in figure 3. In some cases the matching fails to converge to the ideal representation due to excessive deformation or bad initialization. To discard these bad examples, we measure the final distance of image pixels to the lines of final representation. If this distance exceeds some threshold, the image is taken as bad adjusted and it is discarded from the learning set.

With the lineal representation of all sample images, we have, for each line of the symbol, a set of representations corresponding to all the variations of that line found in the learning set. Thus, for each parameter of the line (position, orientation and length), we can build the distribution of the values showing the variability of that parameter along the sample images. From this distributions we can calculate the mean and the standard deviation for each parameter and we can also calculate the covariance matrix among parameters to get interactions among them. In computing the mean of the parameters we cannot directly calculate the mean of the orientation parameter because it could lead to erroneous situations due to the periodicity of orientation. Then, to get the mean of all line orientations we find the angle, α which minimizes the following expression:

$$\sum_{i=1}^{n} \sin^2(\alpha_i - \alpha) \tag{6}$$

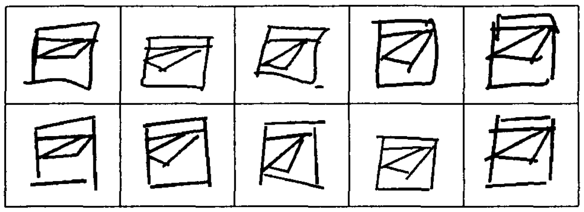

Figure 4. Input images (above) and their lineal representation got by deformable template matching (below).

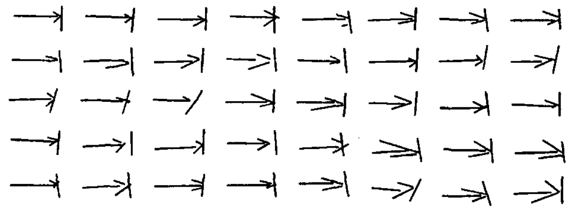

Figure 5. Lineal representation of images used in learning for *arrow* symbol.

where n is the number of sample images and α_i is the orientation of the line for each image.

The information extracted from parameter distributions (mean and standard deviation) allows to create the final representation of the symbol. The mean values for each parameter are used to define the lines composing the ideal model of the symbol, while standard deviation is used to fix the standard deviation in probability distributions employed in the definition of internal energy - expression 2-. In this way, we can infer from the sample set, both the ideal representation of the symbol and the parameters needed to define the cost function.

4. Results and discussion

We have applied this method to the learning of twelve hand-drawn architectural symbols. Our sample set consists of fifty images of each symbol, drawn without any kind of constraint by ten different people. Thus, images show intrinsic variability of hand-drawn symbols, as we can see in figures 4 and 5.

The first step in learning is getting the lineal description of the sample images. We have taken each image and we have applied deformable template matching to match it with its corresponding symbol. Figures 4 and 5 show some examples of matching. They show how the final description of images results in a set of lines accurately representing them.

However, and as we have explained before, deformable template matching is not always able to find a good representation for all images. Bad approximations can be filtered out by removing from the sample set those images with high distance from image pixels to the final lineal representation (figure 3). The resulting set, as that shown in figure 5 for the arrow symbol, is the base to derive the final representation of the symbol.

The final description of every symbol is computed by taking the mean and the standard deviation for each line in the symbol. Applying this step to all twelve symbols, we have got the symbol representations shown in figure 6.

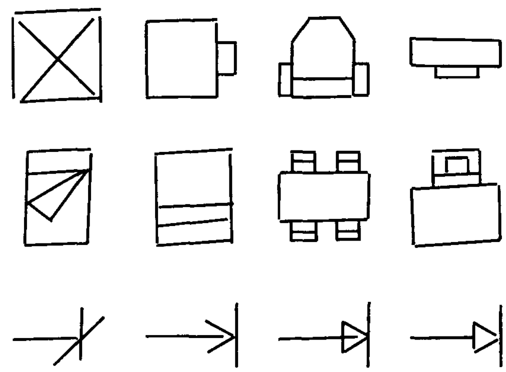

Figure 6. Learned Representation of a set of architectural symbols.

These representations correspond to the mean of position, orientation and length for every line of the symbol, taking all sample images. These final shapes reflect the expected ideal shape of every symbol, but with slight modifications due to hand-drawing distortions which have been learned from the sample set. In figure 7 we can see the distribution of values for the midpoint coordinate x of the four lines of an *arrow* along all sample images. We can see that lines 3 and 4 have higher variability and then, their standard deviation is also higher. This information is added to the symbol representation when setting the standard deviation in the probability distribution defining internal energy - equation 2 -.

5. Conclusions

Accurate description of the shape of symbols taking into account its variability is a key issue in the performance of any structural symbol recognition approach. Automatic learning of this description from a set of sample images is not an straighforward approach. In this work, we have

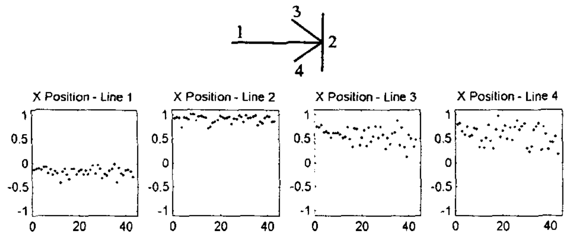

Figure 7. Distribution of midpoint coordinate x for the lines of an *arrow* along the set of samples.

shown one method for automatically getting structural representations of hand-drawn symbols, able to capture the variability due to noise or hand-drawing. We only need to set a fixed number of lines for describing each symbol.

We have used deformable template matching to get a lineal representation of each image. Deformable template matching always finds the description of the symbol with the best approximation to the image, being able to describe even very distorted shapes.

The description of the symbols consists of a set of lines not necessarily connected, each line defined by three parameters: midpoint position, orientation and length. The method doest not only provide a representation of the model, but it also gives information about shape variability by means of the variance of its parameters. This information can be used in recognition to penalize high deformations for lines with low variance while allowing them for lines with high variance. Automatic integration of this information for recognition is an issue to be further investigated.

We also think that it could be very interesting and promising to explore the mapping from this representation to a graph structure, as graphs are the most used representation in structural pattern recognition. This could lead to the extension of this approach to learning graph representations of symbols, allowing both to compute an approximation of the mean graph and an estimation of the cost function employed in error-tolerant graph isomorphism.

References

[1] C. Ah-Soon and K. Tombre. Architectural symbol recognition using a network of constraints. *Pattern Recognition Letters*, 22(2):231–248, February 2001.

[2] L.P. Cordella, P. Foggia, R. Genna, and M. Vento. Prototyping structural descriptions: an inductive learning approach. In A. Amin, D. Dori, P. Pudil, and H. Freeman, editors, *Advances in Pattern Recognition*, pages 339–348. Springer Verlag, Berlin, 1998.

[3] A.K. Jain, Y. Zhong, and M.P. Dubuisson-Jolly. Deformable template models: A review. *Signal Processing*, 71:109–129, 1998.

[4] X. Jiang, A. Mnger, and H. Bunke. Synthesis of representative graphical symbols by computing generalized median graph. In A.K. Chhabra and D. Dori, editors, *Graphics Recognition: Recent Advances*, pages 183–192. Springer Verlag, Berlin, 2000.

[5] S. Lee. Recognizing hand-written electrical circuit symbols with attributed graph matching. In H.S. Baird, H. Bunke, and K. Yamamoto, editors, *Structured Document Analysis*, pages 340–358. Springer Verlag, Berlin, 1992.

[6] J. Lladós, G. Sánchez, and E. Martí. A string-based method to recognize symbols and structural textures in architectural plans. In K. Tombre and A. Chhabra, editors, *Graphics Recognition: Algorithms and Systems, Selected Papers from Second International Workshop on Graphics Recognition, GREC'97*, pages 91–103. Springer, Berlin, 1998. Volume 1389 of Lecture Notes in Computer Science.

[7] B.T. Messmer and H. Bunke. Automatic learning and recognition of graphical symbols in engineering drawings. In Kasturi R and K. Tombre, editors, *Graphics Recognition: Methods and Applications*, pages 123–134. Springer Verlag, Berlin, 1996.

[8] E. Valveny and E. Martí. A bayesian framework for hand-drawn symbol recognition in graphic documents. In *Proceedings of 15th IAPR International Conference on Pattern Recognition, ICPR'00*, volume 2, pages 239–242, Barcelona, Spain, September 2000.