ELSEVIER

Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Pattern Recognition Letters 24 (2003) 2857–2867

Pattern Recognition Letters

www.elsevier.com/locate/patrec

A model for image generation and symbol recognition through the deformation of lineal shapes

E. Valveny [*,1], E. Martí [2]

Computer Vision Center, Dept. Informàtica, Universitat Autònoma de Barcelona, Edifici O, Campus UAB,
08193 Bellaterra (Cerdanyola del Valles), Spain

Received 17 October 2002; received in revised form 19 May 2003

Abstract

We describe a general framework for the recognition of distorted images of lineal shapes, which relies on three items: a model to represent lineal shapes and their deformations, a model for the generation of distorted binary images and the combination of both models in a common probabilistic framework, where the generation of deformations is related to an internal energy, and the generation of binary images to an external energy. Then, recognition consists in the minimization of a global energy function, performed by using the EM algorithm. This general framework has been applied to the recognition of hand-drawn lineal symbols in graphic documents.

Keywords: Symbol recognition; Graphics recognition; Document analysis; Deformable models; EM algorithm

1. Introduction

Symbol recognition plays an important role in many graphics recognition systems. For many years, one of the main interests in graphics recognition has been the automatic conversion of the huge amount of existing paper documents, such as maps, architectural plans, engineering drawings, any kind of diagrams, etc. In such systems, identifying and recognizing structured entities other than lines and points is essential in order to achieve intelligent and complete interpretation of the document. This has been the motivation of much research in graphic symbol recognition until now. Another potential set of applications of symbol recognition is the development of user-friendly interfaces based on recognizing hand-drawn input from the user. Recently, thanks to the expansion of pen-based interfaces, this kind of applications are receiving increasing attention (Landay and Myers, 2001; Pimentel et al., 2001; Valois et al., 2001; Wenyin et al., 2001). Robustness to shape variability and distortion is one of the challenging points to achieve good performance.

Several reviews of the state of the art on symbol recognition have been published in the last editions

[*] Corresponding author. Tel.: +34935811863; fax: +34935811670.

E-mail addresses: ernest@cvc.uab.es (E. Valveny), enric@cvc.uab.es (E. Martí).

[1] Partially supported by CICYT TIC2000-0382, Spain.

[2] Partially supported by CICYT TIC2000-1635-C04-04 and CICYT TIC2000-0399-C02-01, Spain.

0167-8655/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S0167-8655(03)00144-2

of the *International Workshop on Graphics Recognition* (Chhabra, 1998; Cordella and Vento, 2000; Lladós et al., 2002). Both structural and statistical pattern recognition methods have been used when dealing with shape variability and noise. Structural pattern recognition has been widely used in symbol recognition since symbols are easily described with structural representations, such as graphs (Lee, 1992; Lladós et al., 2001; Messmer and Bunke, 1998) or grammars (Sánchez et al., 2001). Distortion can be handled using error-tolerant matching between the image and the model. Other structural approaches are based on the propagation of geometric and topological constraints (Ah-Soon and Tombre, 2001). Statistical pattern recognition approaches for symbol recognition are usually based on extracting geometric features from the image and applying some classifier such as the NN-neighbours rule, Bayes' classifiers or decision trees (Pimentel et al., 2001; Samet and Soffer, 1996). However, its performance when shape variability increases depends very much on selecting a good set of features, which is not always possible.

In this work, we will address the problem of symbol recognition under conditions of noise and unrestricted shape variability describing a method based on deformable models and a probabilistic framework, similar to the approach described in (Revow et al., 1996) for character recognition. In our approach, however, we will take advantage of lineal structure of graphic symbols to define the representation of symbols and the generation of deformations. We argue that deformable template matching is flexible enough to handle the kind of distortions produced by hand-drawing. Deformable models have been successfully applied to a wide range of computer vision applications (Jain et al., 1998). However, only few applications of elastic matching can be found (Burr, 1981; Pavlidis et al., 1998) for graphics recognition. In deformable template matching a model of the object is deformed in order to fit the input image, finding the deformation which minimizes a global energy function, composed of internal energy—a measure of the distortion from original shape—and external energy—a measure of similarity to input image.

The paper is organized as follows. In Section 2, we define the probability of generation for an input image. Section 3 describes symbol representation and how to get symbol deformations and their probability. Then, in Section 4 both probabilities are combined to define a global framework for symbol recognition. In Section 5, some results of the application of this framework are presented and finally, in Section 6, we draw conclusions from this work.

2. Image generation

This section is concerned with the generation of binary images corresponding to hand-drawn lineal symbols. We will describe a framework which associates a probability of generation to every pixel from the lineal model of a symbol and thus, an overall probability for the whole image, which can be used to define the similarity between the image and the symbol.

Graphic symbols are composed of straight lines. However, hand-drawing, scanning and thinning to one-pixel width introduces noise and distortion in the shape of lines. Therefore, image pixels will not follow a perfect lineal path along the expected line location. All around every line, we can define an uncertainty area where pixels can be generated. We will describe this area by a gaussian distribution, where the probability of generation associated to every pixel depends on the distance between the pixel and the line, noted as d_p, but also on the angle difference between them, noted as d_α. Thus, pixels generated by a line are supposed to be close to it and to have similar orientation.

Position distance d_p between a pixel p and a segment s is computed, depending on the projection of the pixel onto the line defining the segment, as the minimum distance between the pixel and the line (Fig. 1a) or as the distance between the pixel and the closest end-point (Fig. 1b).

Angle difference d_α is computed taking the sin of the difference in orientation between the pixel and the segment. The use of sin makes d_α independent of periodic differences. The orientation of a pixel is the tangent angle of the image skeleton at that point, computed from its neighbor pixels (Fig. 1c).

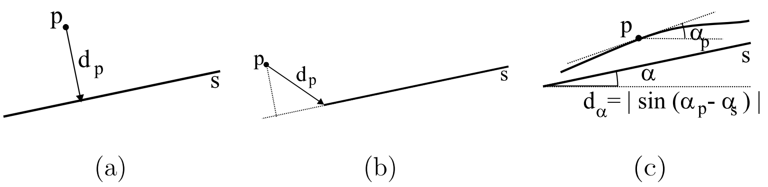


Fig. 1. Distance between a point and a line.

Then, position distance and angle difference are combined to define a global distance measure $d(p,s)$. We have decided to take d_p as the basis for global distance, and to weigh it by a factor depending on d_α and k_α, a constant parameter which allows to control the influence of angle difference in the distance (Fig. 2). Two reasons support this decision: on the one hand, visual evidence of major relevance of position distance; on the other hand, pixel orientation cannot be an exact measure due to its local computation and image discretization. Thus, the definition of $d(p,s)$ is:

$$d(p,s) = (1 + k_\alpha \cdot d_\alpha(p,s)) \cdot d_p(p,s) \tag{1}$$

As explained before, the probability of generation of a pixel from a given segment, noted as $P(p|s)$, is defined as a gaussian distribution based on $d(p,s)$:

$$P(p|s) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \mathrm{e}^{-\frac{d^2(p,s)}{2\sigma^2}} \tag{2}$$

where σ is the standard deviation of distribution. It allows to control the size of the area of influence for every line and therefore, the degree of accepted distortion when hand-drawing a line.

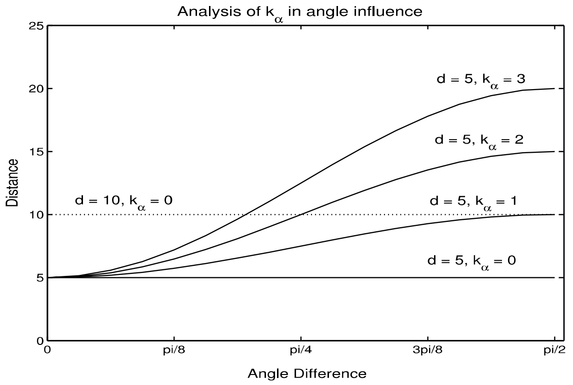


Fig. 2. Analysis of influence of angle difference in the distance.

Every segment acts as a pixel generator. Any pixel can be generated from any of the symbol segments. Then, any pixel has a global probability of generation from the whole symbol, computed by summing up the probabilities of generation from every segment. Moreover, assuming that all pixels of an image have independent probabilities of generation, the probability of generation of a whole image from a given symbol, $P(I|S)$, can be computed by multiplying the probabilities of generation for all pixels. Then, given an image I composed of n pixels, $I = (p_1, \ldots, p_n)$, and a symbol S composed of m segments, $S = (s_1, \ldots, s_m)$, $P(I|S)$, can be expressed as:

$$P(I|S) = \prod_{i=1}^{n} P(p_i|S) = \prod_{i=1}^{n} \sum_{j=1}^{m} P(p_i|s_j) \tag{3}$$

The probability of generation of an image gives a measure of how well it fits the lineal representation of the symbol. In Section 4 it will be used to define a similarity measure between the image and the symbol for matching.

3. Deformation of lineal models

As we are concerned with the recognition of lineal symbols, we can represent the ideal shape of a symbol with a set of straight lines, not necessarily connected. At the same time, we want to recognize hand-drawn symbols. Then, we need some model to generate, from the ideal shape of a symbol, deformations resembling those produced by hand-drawing. In this section we will introduce such a model, based on a probabilistic explanation of deformations, which allows to handle uncertainty in hand-drawing.

A symbol S will then be represented as a set of independent straight segments, $S \equiv \{s_1, \ldots, s_m\}$.

Each segment s is defined with four parameters: mid-point coordinates, orientation and length, $s \equiv (x, y, \alpha, \lambda)$.

With this representation, shape deformations D of the symbol can easily be generated by modification of segment parameters. In fact, any segment can be transformed into any other segment (Fig. 3). However, transformations resulting in not valid shapes of the symbol must be forbidden. As the border between valid and not valid shapes is not clear, we will use a probabilistic model to measure the "validity" of a shape.

Deformations will be generated through the application of independent transformations to every segment. As segments, a transformation ψ is defined by four parameters: the translation of mid-point position Δx and Δy, the change in orientation $\Delta\alpha$ and a scaling factor applied to the segment length $\Delta\lambda$. In this way we can translate, rotate and scale a segment in any desired way. Then, if $\Psi \equiv \{\psi_1, \ldots, \psi_n\}$, is a set of transformations to be applied to every segment, a new deformation D of a symbol S can be generated in the following way:

$$\begin{aligned} D = \Psi(S) &= \{\psi_1(s_1), \ldots, \psi_n(s_n)\} \\ &= \{s'_1, \ldots, s'_n\} \end{aligned} \tag{4}$$

$$s' = \psi(s) = (x + \Delta x, y + \Delta y, \alpha + \Delta\alpha, \lambda \cdot \Delta\lambda) \tag{5}$$

To avoid free symbol deformation, we will associate a probability value to each deformation. This value will measure the certainty for the deformation, of being a valid representation of the symbol. To define this probability we will make two assumptions: first, all transformations applied to every segment and to every parameter are independent from each other; secondly, transformations follow gaussian distributions based on the parameter of the transformation. This way, as the amount of deformation increases, its probability decreases to zero. But at the same time, slight deformations due to hand-drawing can be given high probabilities depending on the value of standard deviation. With these assumptions, the probability associated to a deformation D, denoted $P(D|S)$ is expressed in the following way:

$$\begin{aligned} P(D|S) = P(\Psi|S) &= \prod_{i=1}^{n} P(\psi_i|s_i) = P(\psi|s) \\ &= K \cdot \mathrm{e}^{-\frac{\Delta x^2}{2\sigma_x^2}} \cdot \mathrm{e}^{-\frac{\Delta y^2}{2\sigma_y^2}} \cdot \mathrm{e}^{-\frac{\sin \Delta\theta^2}{2\sigma_\theta^2}} \cdot \mathrm{e}^{-\frac{\Delta\lambda^2}{2\sigma_\lambda^2}} \end{aligned} \tag{6}$$

where σ_x, σ_y, σ_θ and σ_λ are standard deviations for every parameter of local transformation and K is the product of gaussian normalizing factors. Standard deviations can be fixed using the learning procedure described in (Valveny and Martí, 2001) which permits to derive the ideal shape of every symbol and the main variation modes for each line.

We will call this kind of deformations local deformations, because they are achieved through independent transformations locally applied to every segment. They do change the global shape of the symbol. On the other hand, we also need to represent another kind of deformations of a symbol, which we will call global deformations. These deformations will allow to represent global translation, rotation and scaling of the symbol (Fig. 4). They are equally applied to all the lines of the symbol. Therefore, they do not change its global shape. All of them will generate valid representations of the symbol and it is not necessary to associate them any probability measure of "validity".

Global deformations are generated by the application of global transformations. A global transformation Φ is defined by five parameters: translation in x and y, t_x and t_y, angle of rotation, θ and scaling in x and y, s_x and s_y. The application of a transformation Φ to a symbol S, denoted by $\Phi(S)$ produces a deformation D through the successive application of global scaling, rotation and translation to every segment:

$$D = \Phi(S) = \{\Phi(s_1), \ldots, \Phi(s_n)\} = \{s'_1, \ldots, s'_n\} \tag{7}$$

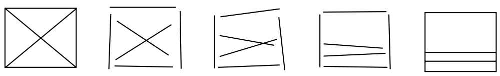


Fig. 3. Example of unrestricted symbol deformation.

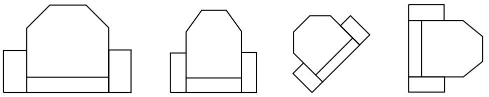


Fig. 4. Example of global deformations.

4. Symbol recognition

Combining the schemes defined in previous sections, the matching of an input image I with a given symbol S will consist in finding the deformation D of the symbol with higher probability of generation for the input image. As deformations are not equally probable, the probability of deformations must also be taken into account. This approach leads to a Bayesian framework where probability of image generation $P(I|D)$ stands for likelihood and probability of deformations $P(D|S)$ stands for prior information. Bayes' rule is used to combine both probabilities getting the posterior probability of a deformation:

$$P(D|I,S) = \frac{P(I|D,S)P(D|S)}{P(I|S)} \tag{8}$$

Then, matching consists in finding the deformation $\widehat{D}$ maximizing $P(D|I,S)$. As we assume that all input images have the same probability $P(I|S)$,

$$\begin{aligned}\widehat{D} &= \arg\max_D P(D|I,S)\\ &= \arg\max_D P(I|D,S)P(D|S)\end{aligned} \tag{9}$$

Taking the negative log of the previous expression, matching can also be viewed as the minimization of an energy function E composed of two terms: internal energy E_{int} and external energy, E_{ext}, which are respectively related to prior probability and likelihood through the following equivalences:

$$E_{\text{int}}(D) = -\log P(D|S) \tag{10}$$

$$E_{\text{ext}}(D) = -\log P(I|D,S) \tag{11}$$

$$\widehat{D} = \arg\min_D E(D) \tag{12}$$

With this interpretation, $\widehat{D}$ can be viewed as the equilibrium point reached after the application of two opposite forces: one trying to keep the shape of the symbol close to the original shape (internal energy), and the other one trying to push it to the shape of the input image (external energy). It will correspond to the deformation which best fits the input image with the lowest degree of distortion from the ideal shape.

Given an input image and a set of predefined symbols $\{S_1,\dots,S_n\}$, we can match the input image with every symbol resulting in a set of optimal deformations $\{\widehat{D}_1,\dots,\widehat{D}_n\}$, one for every symbol. As the final energy value of deformations is related to its posterior probability through equations (9)–(11), input image I will be recognized as symbol S_i where $i = \arg\min_j E(\widehat{D}_j)$.

4.1. Energy minimization

Finding $\widehat{D}$ according to expression (12) requires the minimization of a complex energy function. Generally, this is not a straightforward task because this function has many local minima. However, we can take advantage of the following observation: if we could know which image points have been generated by each line, optimal line parameters could be easily computed by finding the best fitting of a line to this set of points.

This fact leads to the use of the EM algorithm to find the minimum of energy. EM algorithm is well suited in problems with missing or incomplete information. It works by applying iteratively two steps, until convergence is reached. In the expectation step, missing information is estimated. In the maximization step, a partial solution is found assuming that the information estimated in the expectation step is true. In our case, the missing information is the association between lines and points, which can be estimated using the probability of generation, $P(p|s)$, defined in expression (2). In the maximization step, it will be used as a weighting factor to measure the influence of every point when computing the optimal parameters of a line.

Then, in the expectation step, the probability of generation p_{ij} between every point p_i and every line s_j is estimated. It is normalized so that, for each point, it sums up 1 for all lines:

$$p_{ij} = \frac{P(p_i|s_j)}{\sum_{k=1}^{m} P(p_i|s_k)} \quad i \leqslant n, \quad j \leqslant m \tag{13}$$

In the maximization step, as we have assumed that all parameters of segments are independent, computation of optimal parameters can be done separately for every parameter of every segment. Influence of internal energy can also be split up in independent components for every parameter by combining expressions (10) and (6). The contribution of constant factor K can be ignored in the minimization process:

$$E_{\text{int}} = \sum_{i=1}^{n} \frac{\Delta x_i^2}{2\sigma_{\Delta x_i}^2} + \frac{\Delta y_i^2}{2\sigma_{\Delta y_i}^2} + \frac{\sin \Delta\theta_i^2}{2\sigma_{\Delta\theta_i}^2} + \frac{\Delta\lambda_i^2}{2\sigma_{\Delta\lambda_i}^2}$$
$$= \sum_{i=1}^{n} E_{\text{int}}^{x_i} + E_{\text{int}}^{y_i} + E_{\text{int}}^{\theta_i} + E_{\text{int}}^{\lambda_i} \quad (14)$$

Combination of internal and external energy requires a weighting factor α which normalizes the values of both functions to the same range of values. It permits to control the influence of internal energy and thus, the degree of deformation allowed. Empirically, we have set the value of α to 0.001. Then, the global energy function to be minimized is $E = \alpha \cdot E_{\text{int}} + E_{\text{ext}}$.

To find the new representation of the symbol, we have to find the optimal representation of every line which best fits the pixels of the image. In the expectation step we have computed the probability of generation of every pixel by each of the lines. Then, we can take for every line, all pixels having a probability of generation greater than zero, and compute the parameters of the line which best fits this set of points. Every point is weighted by its probability of generation. This way more probable points have greater influence.

The optimal parameters which best fits a segment to a set of points can be computed in three stages: first, we find the orientation of the line minimizing the squared sum of distances; secondly, we compute the center of the segment as the point minimizing the distance to the set of points; and finally, we estimate the optimal length of the segment.

First, optimal orientation will be computed as the orientation of the line minimizing the sum of distances (d_p) from it to the set of points. We have to take into account that each point will contribute to the distance with a weighting factor according to p_{ij}—expression (13)—and the influence of internal energy. Then, the expression to minimize is:

$$E_j^{\theta} = \alpha \cdot E_{\text{int}}^{\theta_j} + \sum_{i=1}^{n} \bar{p}_{ij} \cdot d_p^2(p_i, l_j) \quad (15)$$

where $\bar{p}_{ij}$ is the probability of generation p_{ij}, normalized by the sum of probabilities between all points and segment l_j to sum up 1 for all points.

Analytical minimization of this expression permits an easy computation of optimal orientation, which is used in the following stages of the maximization step. The next stage consists in computing optimal mid-point coordinates of every segment. They can be computed by minimizing the distance of mid-point to the set of points. As before, weighting factors for every point and influence of internal energy must be taken into account, leading to the following expression which is again analytically minimized to get x and y values.

$$E_j^{x,y} = \alpha \cdot E_{\text{int}}^{x_j} + \alpha \cdot E_{\text{int}}^{y_j} + \sum_{i=1}^{n} \bar{p}_{ij}((x_i - x_j)^2 + (y_i - y_j)^2) \quad (16)$$

The final stage of the maximization step is computation of optimal segment length, where we must take some approximations. We observe that the sum of the distances from all points to the mid-point of the segment is an approximation of half of the segment length. Then, adding to this fact the influence of internal energy, the global energy function to minimize is the following:

$$E_j^{\lambda} = \alpha \cdot E_{\text{int}}^{\lambda_i} + \left(\sum_{i=1}^{n} \bar{p}_{ij} |d(p_i, p_j)| - \frac{\lambda}{2} \right)^2 \quad (17)$$

Once orientation, mid-point coordinates and length have been computed for all segments, we have got a new representation of the symbol which fits the input image better. Then, expectation and maximization steps are iterated until convergence is reached. At each iteration standard deviation used in computing p_{ij} is decreased in order to achieve better approximations. Criteria for convergence and standard deviation decreasing are discussed in Section 5.1.

5. Results and discussion

The origin of this work was the study of new man–machine interfaces for input of graphic

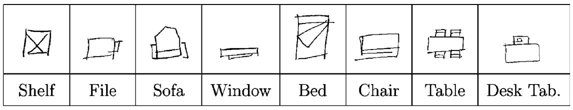


Fig. 5. One sample image of every symbol.

information in architectural environments based on hand-drawn user input. Then, we have used a set of usual symbols in architectural drawings to test our approach. These symbols describe furniture elements in a building, such as tables, chairs, windows, etc. We have selected eight different symbols, taking fifty images of each of them, drawn by ten different people in a totally unconstrained way. These images are representative of a great number of symbol distortions. Several examples are shown in Fig. 5.

We will analyze two different issues of the application of the general framework described in previous sections to this set of sample images. First, we will discuss the criteria for convergence of energy minimization procedure. Secondly, we will focus our attention on experiments about performance measures in terms of recognition rates and computational complexity.

5.1. Convergence criteria

As described in Section 4.1, matching relies on the minimization of an energy function. This function depends on many parameters and thus, its minimization is not straightforward. To avoid local minima, good initialization and convergence criteria must be chosen.

Initialization is required in order to reduce the distance between initial and minimal models, so that EM algorithm could find the path between them. We can do that thanks to global transformations introduced in Section 3, which let the model change without any cost, applying global rotations, scaling and translation to all the lines. Then, we can find the global transformation which minimizes the distance from the model to the input image using the initialization step described in (Valveny and Martí, 2000). Fig. 6 shows the difference in approximation for a given image, without and with a previous initialization step.

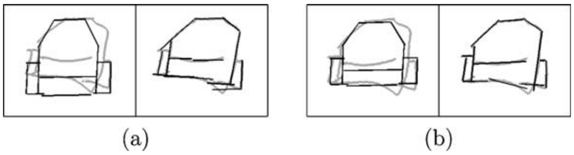
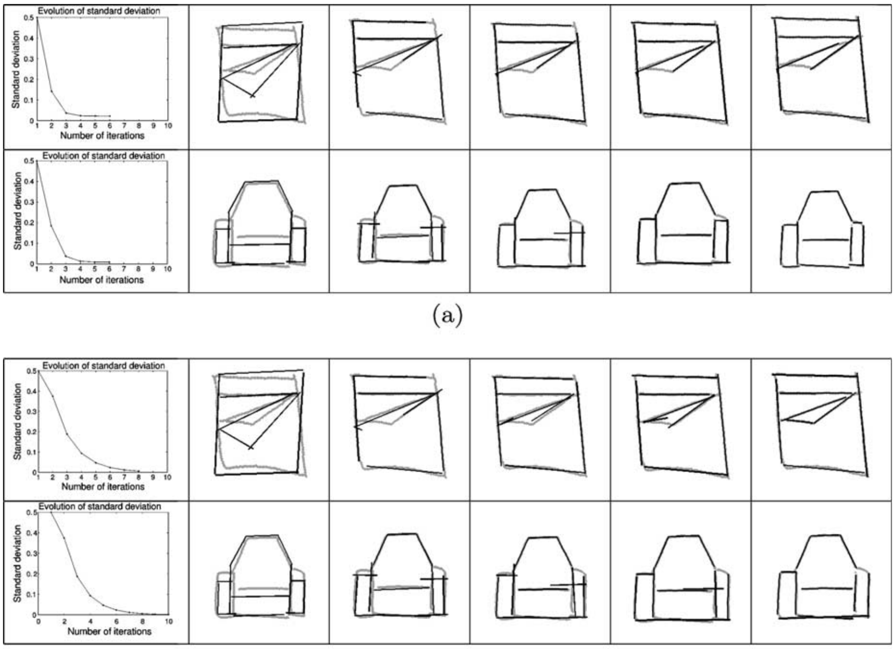


Fig. 6. (a) Matching without initialization. (b) Matching with initialization using global transformations.

Convergence of the EM algorithm to the global minimum depends basically on how we update, at each iteration, the standard deviation used in the expectation step. This standard deviation controls the area of influence for each line in the model. The higher its value, the greater will the line be attracted by farther pixels. Thus, it is clear that, at the beginning, when the model is slightly or not deformed, standard deviation must be high in order to let the lines move towards the most appropriate pixels, which can be far away from the line. However, as the minimization process goes forward and deformations are closer to the image, standard deviation must be low in order to reduce the area of influence for each line only to those pixels corresponding to it.

The problem lies in the decreasing rate of standard deviation. If it decreases too fast, we lose ability to fit lines which are initially far away from their corresponding pixels. On the other hand, if decreasing rate is too slow, some lines may be confused by neighboring wrong pixels and may fail to converge. Fig. 7 shows this fact. First, we have used a variable decreasing criterion. Standard deviation is updated, at each iteration, taking the value of the distance between every pixel and every line of the current deformation, as defined in Eq. (1). This results in a quick decrease of standard deviation. Secondly, we have used a fixed updating criterion, setting standard deviation to half its last value. This results in a smoother decreasing curve, which permits the recovery of some wrong adjusted line in the first symbol. However, in the second symbol, lines are confused during the first iteration by nearby pixels, and the final result is worse than with the first criterion.

Analyzing these experimental results, we have decided to use the first decreasing criterion, because, with similar matching results, it has lower



(a)

(b)

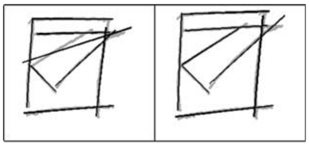
Fig. 7. Evolution of standard deviation in e-step. (a) Variable updating criterion. (b) Fixed updating criterion.

computation time, as it usually converges more quickly. We let the minimization process run until the difference between two successive standard deviations is small enough, or until we reach a fixed number of iterations. Table 1 shows the average number of iterations needed when matching every symbol. Symbols with more lines are symbols with greater average number of iterations.

5.2. Performance evaluation

In this section we will discuss the performance of our approach in terms of recognition rates and computational complexity. The first issue to consider is the definition of a model for every symbol. In (Valveny and Martí, 2001) we have described a procedure to learn the representation of a symbol and the parameters for internal energy from a set of sample images. We have applied this procedure and we can see in Fig. 8, how matching improves with new symbol representation.

Table 2 shows global recognition rates for images of each symbol. We get an overall recognition rate of 97.75% for all images. To test the signifi-



(a) (b)

Fig. 8. (a) Matching before learning. (b) Matching after learning.

Table 1
Average number of iterations for every symbol

Shelf	File	Sofa	Window	Bed	Chair	Table	Desk tab.	Total
5.4	5.7	6.26	5.22	5.56	5.82	6.3	6.1	5.795

Table 2
Recognition rates

Shelf 	File 	Sofa 	Window 	Bed 	Chair 	Table 	Desk tab. 	Total
100%	98%	100%	86%	100%	98%	100%	100%	97.75%

cance of this recognition rate with this set of deformed images we have compared it with the recognition rate achieved by a classification method based on a set of standard and usual geometric features (area, major and minor axis length, eccentricity, euler number, etc.) provided by the Matlab package. With this method, we only get an overall recognition rate of 50.25%, due to the variability of images.

As explained in Section 4, recognition is carried out by matching an image with all symbols and taking the symbol with the lowest final energy. Exact visual matching between an image and the symbol is not needed in order to get correct recognition. What is important is final energy value in relation with other symbols. We can see, in Fig. 9, how matching for images of the symbol *table* is not always very precise because of the great amount of lines in the symbol and the proximity among them. However it is better than matching with any other symbol, yielding 100% of recognition. In Fig. 10, final energy values for the matching of all images of the symbol *table* with their model with the model resulting in the second best matching for every image are displayed.

The symbol with lowest recognition rates is the symbol *window*. Confusions come from the similarity of this symbol with the symbol *chair*. Thus, deformations of the symbol *chair* can be generated, which match images of the symbol *window*. Fig. 11 illustrates this fact. In it, we can see for each image of the symbol *window*, the final

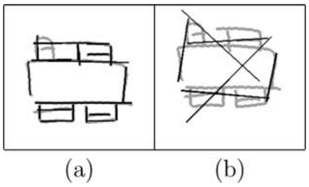


Fig. 9. (a) Matching with the symbol *table*. (b) Matching with the symbol *shelf*.

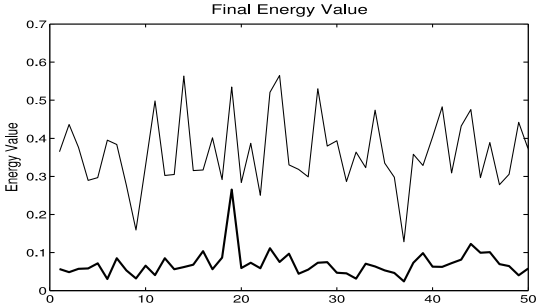
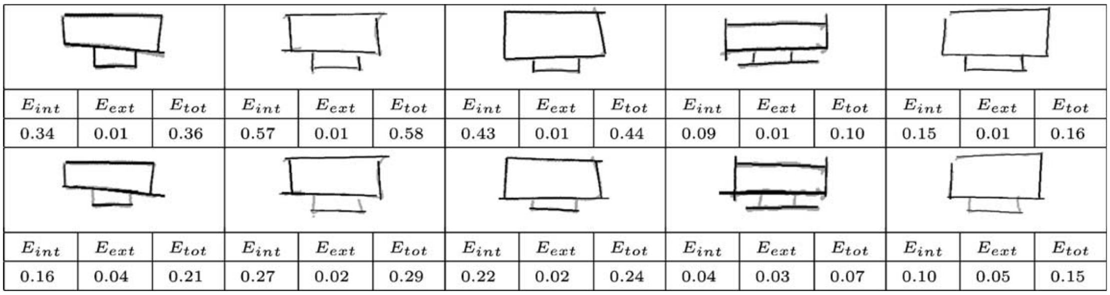


Fig. 10. Final energy value for best two matchings for images of symbol *table*.



E_{int}	E_{ext}	E_{tot}	E_{int}	E_{ext}	E_{tot}	E_{int}	E_{ext}	E_{tot}	E_{int}	E_{ext}	E_{tot}	E_{int}	E_{ext}	E_{tot}
0.34	0.01	0.36	0.57	0.01	0.58	0.43	0.01	0.44	0.09	0.01	0.10	0.15	0.01	0.16
E_{int}	E_{ext}	E_{tot}	E_{int}	E_{ext}	E_{tot}	E_{int}	E_{ext}	E_{tot}	E_{int}	E_{ext}	E_{tot}	E_{int}	E_{ext}	E_{tot}
0.16	0.04	0.21	0.27	0.02	0.29	0.22	0.02	0.24	0.04	0.03	0.07	0.10	0.05	0.15

Fig. 11. Matching of not recognized images of symbol *window*. Above: matching with symbol *window*. Below: matching with symbol *chair*.

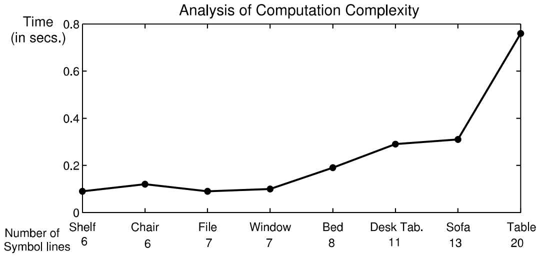


Fig. 12. Analysis of computation time as the number of lines in the symbol increases.

matching with the symbol *window* and with the symbol *chair* with their respective energy values. Although external energy values are similar in most cases, more variability is allowed in the symbol *chair* and therefore, internal energy values are lower for it.

Finally, in Fig. 12 we analyze the computation time of the algorithm. It shows the average time of matching all sample images with each symbol. We can see how computational complexity is approximately linear in the number of lines of the symbol.

6. Conclusions and future work

Tolerance to shape variability is a central issue in symbol recognition for the purpose of designing user interfaces taking advantage of hand-drawing, such as in pen-based systems. In this work we have introduced a general probabilistic framework based on deformable template matching which has shown itself to be well-suited to handle uncertainty in the shape of hand-drawn symbols. We have taken advantage of the lineal shape of the symbols to define a deformation model based on a lineal representation and the application of transformations to the lines. Similarity of an input image to a model has also been described in a probabilistic way, defining a probability of generation of the whole image from a given lineal shape, which permits the representation of the lack of precision in hand-drawing. The combination of both models, the deformation and the generation model leads to the definition of an energy function which has to be minimized to match the input image with the symbol. The use of a probabilistic framework permits the carrying out of the minimization of the energy function with the EM algorithm, which simplifies the problem.

We have applied the method to the recognition of unconstrained hand-drawn architectural symbols, with an overall recognition rate of 97.75%. Only similar symbols get confused, due to different variance in deformation parameters. Computation time is approximately linear with the number of lines in the symbol. Although linear, computation time is one of the issues to be improved in order to be able to use the method in applications where the number of different symbols is significant, as the input image has to be matched with the model of all symbols. Approaches reducing the number of points needed to match the image with the symbol or reducing the number of matches through some kind of hierarchical matching of symbol could be explored.

Confusions between similar symbols require better estimation of deformation parameters. In this way, covariance between different parameters could be introduced to better describe interrelations among different lines. Applying geometric or topological restrictions to deformations could also be an alternative to avoid excessive deformations of symbols.

References

Ah-Soon, C., Tombre, K., 2001. Architectural symbol recognition using a network of constraints. PRL 22, 231–248.

Burr, D.J., 1981. Elastic matching of line drawings. IEEE Trans. Pattern Anal. Machine Intell. 3 (6), 708–713.

Chhabra, A., 1998. Graphic symbol recognition: An overview. In: Tombre, K., Chhabra, A., (Eds.), Graphics Recognition: Algorithms and Systems. Springer, Berlin, Vol. 1389 of Lecture Notes in Computer Science, pp. 68–79.

Cordella, L.P., Vento, M., 2000. Symbol and shape recognition. In: Chhabra, A.K., Dori, D. (Eds.), Graphics Recognition: Recent Advances. Springer-Verlag, Berlin, Vol. 1941 of Lecture Notes in Computer Science, pp. 167–182.

Jain, A.K., Zhong, Y., Dubuisson-Jolly, M.P., 1998. Deformable template models: A review. Signal Process. 71, 109–129.

Landay, J.A., Myers, B.A., 2001. Sketching interfaces: Toward more human interface design. IEEE Comput. 34 (3), 56–64.

Lee, S., 1992. Recognizing hand-drawn electrical circuit symbols with attributed graph matching. In: Baird, H.S., Bunke, H., Yamamoto, K. (Eds.), Structured Document Analysis. Springer-Verlag, Berlin, pp. 340–358.

Lladós, J., Martí, E., Villanueva, J.J., 2001. Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. IEEE Trans. Pattern Anal. Machine Intell. 23 (10), 1137–1143.

Lladós, J., Valveny, E., Sánchez, G., Martí, E., 2002. Symbol recognition: Current advances and perspectives. In: Blostein, D., Kwon, T.B. (Eds.), Graphics Recognition: Algorithms and Applications. Springer, Berlin, Vol. 2390 of Lecture Notes in Computer Science, pp. 99–121.

Messmer, B.T., Bunke, H., 1998. A new algorithm for error-tolerant subgraph isomorphism detection. IEEE Trans. Pattern Anal. Machine Intell. 20 (5), 493–503.

Pavlidis, I., Singh, R., Papanikolopoulos, N.P., 1998. On-line handwriting recognition using physics-based shape metamorphosis. Pattern Recognit. 31 (11), 1589–1600.

Pimentel, C.F., Fonseca, M.J., Jorge, J., 2001. Experimental evaluation of a trainable scribble recognizer for caligraphic interfaces. In: Proceedings of 4th International Workshop on Graphics Recognition, September 2001. Kingston, Canada, pp. 85–94.

Revow, M., Williams, C.K.I., Hinton, G., 1996. Using generative models for handwritten digit recognition. IEEE Trans. Pattern Anal. Machine Intell. 18 (6), 592–606.

Samet, H., Soffer, A., 1996. Marco: Map retrieval by content. IEEE Trans. Pattern Anal. Machine Intell. 18 (8), 783–797.

Sánchez, G., Lladós, J., Tombre, K., 2001. An error-correction graph grammar to recognize textured symbols. In: Proceedings of 4th International Workshop on Graphics Recognition, September 2001. Kingston, Canada, pp. 135–146.

Valois, J.P., Côté, M., Cheriet, M., 2001. Online recognition of sketched electrical diagrams. In: Proceedings of 6th International Conference on Document Analysis and Recognition, September 2001. Seattle, USA, pp. 460–464.

Valveny, E., Martí, E., 2000. Deformable template matching within a bayesian framework for hand-written graphic symbol recognition. In: Chhabra, A.K., Dori, D., (Eds.), Graphics Recognition: Recent Advances. Springer. Berlin. Vol. 1941 of Lecture Notes in Computer Science, pp. 193–208.

Valveny, E., Martí, E., 2001. Learning of structural descriptions of graphic symbols using deformable template matching. In: Proceedings of 6th International Conference on Document Analysis and Recognition, September 2001. Seattle, USA, pp. 455–459.

Wenyin, L., Jin, X., Qian, W., Sun, Z., 2001. Inputing composite graphic objects by sketching a few constituent simple shapes. In: Proceedings of 4th IAPR Work on Graphics Recognition, September 2001. Kingston, Canada, pp. 73–84.