Camera-Based Graphical Symbol Detection

Marçal Rusiñol, Josep Lladós
Computer Vision Center
Dept. Ciències de la Computació
Edifici O, UAB, 08193 Bellaterra, Spain
{Marcal, Josep}@cvc.uab.es

Philippe Dosch
Loria UMR 7503
615, rue du jardin botanique, B.P. 101
54602 Villers-lès-Nancy Cedex, France
Philippe.Dosch@loria.fr

Abstract

In this paper we present a method to locate and recognize graphical symbols appearing in real images. A vectorial signature is defined to describe graphical symbols. It is formulated in terms of accumulated length and angular information computed from polygonal approximation of contours. The proposed method aims to locate and recognize graphical symbols in cluttered environments at the same time, without needing a segmentation step. The symbol signature is tolerant to rotation, scale, translation and to distortions such as weak perspective, blurring effect and illumination changes usually present when working with scenes acquired with low resolution cameras in open environments.

1. Introduction

The present resolution of digital cameras integrated on cell phones or PDA devices makes camera-based document analysis viable, and interesting applications are emerging. Driving assistance, blind person aid systems, image retrieval and robotics are a few examples of possible applications. In all these use cases, text and graphics detection and recognition appear as one of the central problems to solve.

In the last years a growing attention has been given to text detection in real environments, and promising results can be appreciated in [5]. Text recognition has been a topic of interest since the dawn of document analysis and commercial OCR applications are offering good enough recognition rates. Nowadays, combining both text detection and text recognition, *i.e.* an application able to "read" not a scanned document but a real image, is feasible.

Contrary to text, graphical symbols, which are usually designed to guide passengers and pedestrians through transportation facilities and other sites of international exchange, can address a universal communication need. Graphics recognition is also a wide topic of interest in the document analysis field. A good review on shape representation and description techniques can be found in [11]. But, even if most of these techniques achieve great recognition rates, they are only applicable to recognize pre-segmented shapes. We consider that *graphics detection in real environments* is still an unexplored interesting research topic. A compact representation of expressive features –signature– combined with a voting scheme, is the used strategy when trying to do what is known as *symbol spotting*. In this paper we study the problem of detection of zones of interest in real images having a high probability to find the queried graphical symbol. Symbol spotting approaches arise to face up the paradigm of recognition and segmentation: *Do we need to recognize for segmenting or do we need to segment to recognize?* Symbol spotting tries to detect symbols while recognizing them at the same time. Obviously, this kind of techniques will achieve less recognition rates than shape descriptors working with pre-segmented shapes.

Most of shape descriptors used for object detection and recognition are based on pixel features. Pixel primitives entail robustness to different degradations and distortions, but also have a high complexity. But, as we can appreciate in Fig. 1, graphical symbols are composed of synthetic simple geometric shapes usually drawn on an uniform background. In that case a vectorial signature involving a lower complexity and coping with geometric information expressed in terms of length and angles is a suitable approach.

Figure 1. Different Information Symbols Present in Airports.

To detect symbols appearing in real images we use a similar approach to Stein and Medioni's method [9], where a polygonal approximation of the contours of a real image is computed. The relevant features to discriminate graphical symbols taken into account are the angles and lengths of the chains of adjacent segments –polylines–. However, when working with vectorial data, some drawbacks have to be faced. The invariance to the number of segments composing a polyline, the invariance to geometric transformations, and the robustness to distortions must be guaranteed to develop useful applications.

The remainder of this paper is organized as follows: we introduce in the next Section how the graphical symbols are represented and how we model the discriminative signature. In Section 3, the symbol spotting process, composed basically of the signature similarity measure and a voting scheme, is presented. We provide the experimental results in Section 4. Finally conclusions are presented in Section 5.

2. Vectorial Signature for Symbol Detection

Graphical symbols found in real environments are usually filled closed shapes drawn on an uniform background, as we have seen in Fig. 1. They are composed of few regions having closed contours. We describe then a symbol as the combination of several closed contours, approximated by the corresponding set of polylines.

Let us further detail how the symbols are represented and encoded in a suitable way to be compared afterwards and how the vectorial signature to discriminate polylines is defined.

2.1. Symbol Representation

A symbol is described by a vectorial signature in terms of basic primitives extracted from image contours. We extract the contours of the image with a Canny edge operator. Then, a polygonal approximation of these contours is computed using the Rosin and West algorithm [7]. Afterwards, each chain of segments resulting of the polygonal approximation is grouped as a polyline instance. These polylines are the features which describe the symbol to be recognized.

Formally, let $P = \{s_1 ... s_n\}$ be a polyline composed of n segments s_i. Each segment s_i is attributed with the tuple $(l_i; \phi_i)$, where l_i denotes the length of the segment s_i and ϕ_i denotes the angle between s_i and s_{i-1} in the counterclockwise direction. A symbol is represented in terms of its building p polylines and denoted as $S = \{P_1 ... P_p\}$. We can see an example of this symbol representation in Fig. 2.

2.2. Proposed Signature Model

One of the major problems of working with vector based features is that in presence of noise or distortion, the raster-to-vector process can result in very different segment chains. In order to be invariant to the number of segments

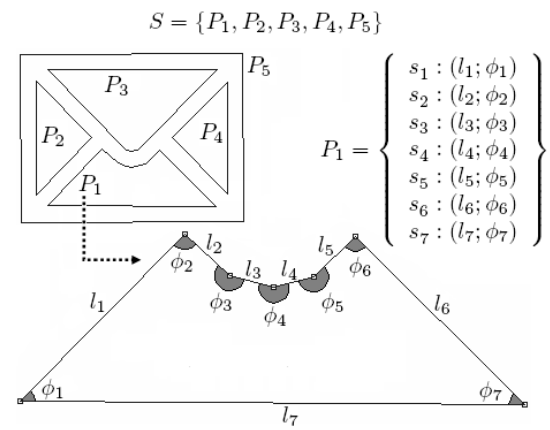

Figure 2. Length and Angle Symbol Representation.

composing a polyline, the vectorial signature is defined in terms of accumulated measures.

Given a polyline P with n segments, a vector of accumulated lengths ℓ normalized by the total length $|P|$, and a vector of accumulated angles Θ, are computed:

$$\begin{array}{l} \ell(i) = \frac{1}{|P|} \times \sum_{k=1}^{i} l_k \\ \Theta(i) = \sum_{k=1}^{i} \phi_k \end{array} \quad where\ 1 \leq i \leq n \qquad (1)$$

Let us then define a mapping function $m(\ell(i)) = \Theta(i)$ assigning the corresponding accumulated angle at each value of $\ell(i)$. As we work with closed contours $m(0) = 0$ and $m(1) = 2\pi$. However $m(x)$ function can not be used as symbol signature as it is, since the use of accumulated measures provokes dependence of the first segment choice.

To guarantee invariance to rotation the idea is to compare the $m(x)$ functions to the accumulated length and angle distributions of an ideal circle, denoted as $c(x)$ and defined as:

$$c(x) = 2\pi x \quad \forall x \in [0,\ 1] \qquad (2)$$

Then, we define the symbol signature $f(x)$ as follows:

$$f(x) = m(x) - c(x) \quad \forall x \in [0,\ 1] \qquad (3)$$

The analysis of the differences between length and accumulated angles of a closed shape and an ideal circle, gives information about the convexities and concavities of the analyzed polyline, and also guarantees invariance to rotation and independence of the reference segment choice.

As we can see in Fig. 3, the two shapes to compare are rotated, so the polygonal approximation step establishes two different starting segments. The use of accumulated

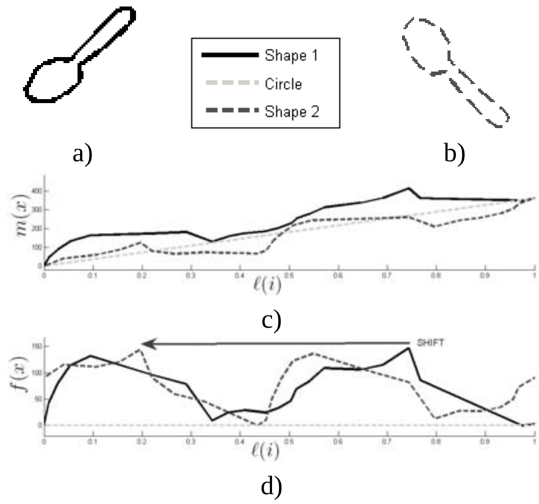

Figure 3. Vectorial Signature. (a) and (b) Two Rotated Versions of the Same Shape. (c) $m(x)$ Length and Accumulated Angles Signal. (d) $f(x)$ Shifted Signatures Due to Rotation.

measures provokes that the two $m(x)$ functions achieve different angle values at a certain covered length. When we compute the $f(x)$ function in relation of the theoretical angle evolution of a circle, we keep the advantages of the accumulated metrics (invariance to scale and number of segments forming the polyline) and the two signatures are comparable since they are only shifted in the x axis.

This description technique copes with a similar geometric information to the well known Curvature Scale Space descriptor proposed in [6]. CSS describes shapes in terms of the expressive curvature inflection points which is the same information we extract without the need to compute the whole scale space.

If the similarity measure between signatures can cope with cyclic shifts, the invariance to the choice of the starting segment and rotation is guaranteed. Let us further describe in the next section how the similarity measure is defined and how the spotting process is done.

3. Symbol Spotting

The proposed symbol spotting process is based on the presence of the polylines composing a symbol in a given region of an image. In this section we introduce the similarity measure between vectorial signatures and the voting scheme, which forms clusters of polylines to segment the regions of interest.

3.1. Similarity measure

A suitable similarity measure between vectorial signatures has to be defined. We need a similarity measure which is able to cope with circular shifts and which consider the trend of the feature vectors.

A normalized cross correlation [3] is often used as a template matching measure. Let $g(x)$ be the signature of the model polyline which we want to find between all the polyline signatures $f(x)$ appearing in the image. The normalized cross correlation between $f(x)$ and $g(x)$ is computed as:

$$f \circ g(x) = \int_{-\infty}^{+\infty} f(\alpha) g(x+\alpha) d\alpha \qquad (4)$$

Since the $f(x)$ vectorial signatures of cyclic shapes start and end at the same values, to consider $f(x)$ as a cyclic signal, we define $F(x)$ as the concatenation of several $f(x)$. The resulting cross correlation between the template $g(x)$ all over $F(x)$ allow us to achieve the invariance to the first segment choice and consequently to rotation. The similarity measure is then defined as:

$$d = \max(F \circ g(x)) \qquad (5)$$

Using the two shapes shown in Fig. 3, we can appreciate in Fig. 4 how the use of the proposed vectorial signature and the similarity measure allow to find a match between $F(x)$ and $g(x)$, assigning a high similarity measure between both shapes.

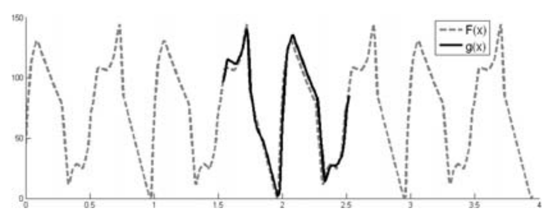

Figure 4. Matching $g(x)$ Over $F(x)$ Using the Normalized Cross Correlation.

3.2. Voting Scheme

Voting schemes are a common strategy when trying to segment and recognize at the same time. Geometric Hashing [10] or the Generalized Hough Transform [1] are well known examples of how this kind of methods work. We can find in the literature several approaches which combine a compact feature representation and voting schemes, used for example for image retrieval applications [4] or for symbol spotting [8] purposes. Since a symbol is composed of

Table 1. Achieved Recognition Rates of Different Shape Descriptors Using the MPEG-7 Database, Extracted from [2].

Descriptor	**CSS**	**Wavelet**	**Eigenvectors**
Rec. Rate	75.44	67.76	70.33
Descriptor	**Zernike**	**DAG**	**Our Method**
Rec. Rate	70.22	60	64.29

several polylines, the zones where a symbol is present receive a high number of votes, forming clusters allowing to segment these zones. Let us detail how we proceed to formulate the voting scheme.

Once the similarity measure between polylines has been formulated, we need to use a voting scheme to cluster the zones of an image where there is presence of the different polylines composing a given symbol. Each image is divided on a grid partition and each polyline appearing in the image cast a vote on its neighboring bins, proportionally to the similarity measure to the closest model polyline. The bins accumulating the higher number of votes are the zones of interest of the image where it is more probable to find the queried symbol. The polylines falling into these activated bins and having a good enough similarity measure are considered as being part of the queried symbol.

The use of voting schemes allow to reject isolated polylines similar to a part of the queried symbol but not surrounded by the rest of parts of the symbol. On the other hand, if one of the polylines composing a symbol is so distorted that the similarity measure miss it, the rest of polylines composing it still contribute to attach importance to the zone of interest, and the symbol can be spotted in despite of the missed part. Let us see in the next Section the experimental results.

4. Experimental Results

The first test was designed to show the discrimination ability of the vectorial signature. To test the method we use the shape images used for the MPEG-7 core experiment, described in [2]. This database consist of 1400 isolated contour shapes grouped in 70 different classes. We can see in Table 1 the recognition results of different shape descriptors over this shape database. Of course, pixel-based descriptors perform better recognition results, but they are also more time consuming and only work with pre-segmented shapes. Studying the recognition results of our method, we can appreciate that the vectorial signature can discriminate well shapes between them and tolerate rotation and scale changes, but do not support the symmetry transformations.

As we tested that the proposed signature is discriminative enough for symbol spotting purposes, for the next two experiments real images are acquired with a low-resolution VGA camera[1] integrated on a cell-phone. The second experiment is focused on a single sign appearing on different environments. We built a groundtruth database with 122 images of the wheelchair symbol. We can see in Fig. 5 the spotting results. The symbol is not well spotted in only 5 images of the whole database. In Fig. 5(a), 5(b) and 5(c) we can see that the method is invariant to scale. In Fig. 5(d) we can see that the method is tolerant to weak perspective. Finally, note in Fig. 5(e) that some false positives appear, as we can appreciate in the zoomed image, all the U letters appearing in the image are confused with the polyline representing the chair. However, the presence of these false positives is expected since the shape of the letter and the symbol are relatively similar.

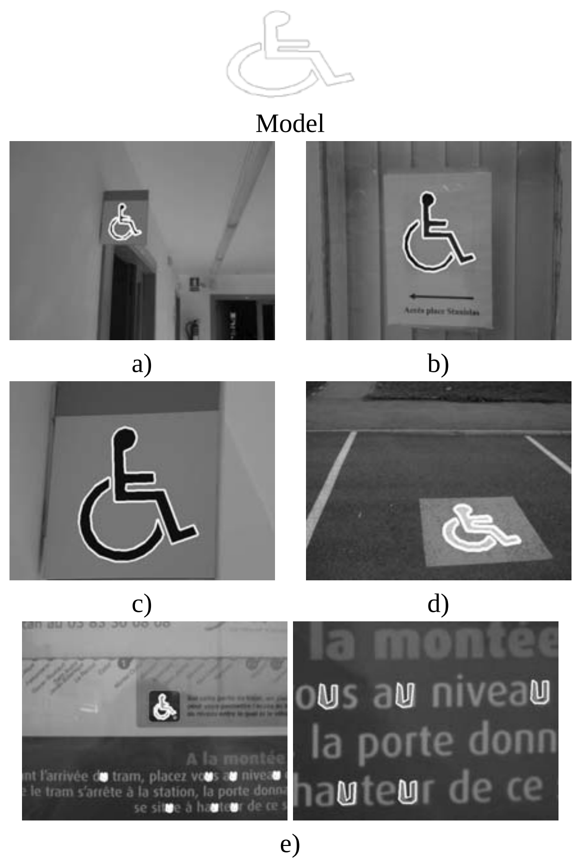

Figure 5. Wheelchair Symbol Spotting in Different Environments.

Finally we tested the method on diverse symbol designs to check the robustness of the method independently of the

[1] 640×480 pixels

queried symbol. In Fig. 6 we can see that acceptable results are achieved when testing the method with different traffic signs.

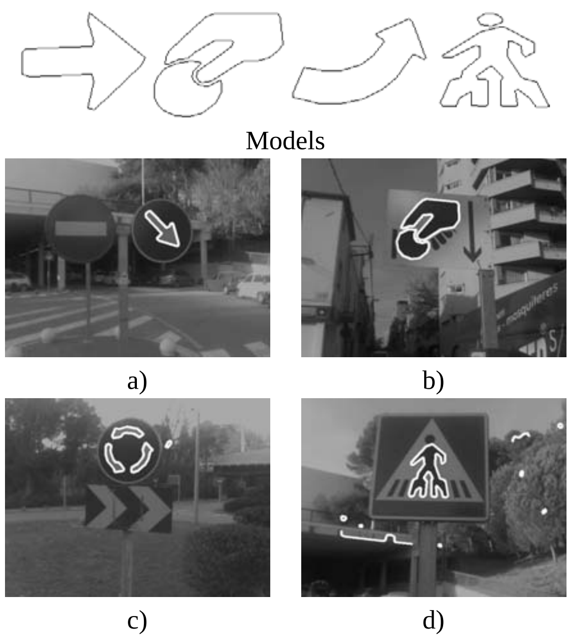

Figure 6. Other Examples of Symbol Spotting in Cluttered Environments.

Studying these results we can conclude that the presence of false positives varies depending on the complexity of the queried shape (the more complex is the shape to find, the less false positives appear) and the complexity of the environment (the more cluttered is the environment, the more false positives appear). These tests show the good location and recognition performance of the method but also reveal some drawbacks of the method. The proposed method is not able to recognize symbols if the contour has been broken since the polylines are also broken and can not be compared.

5. Conclusions

In this paper we have presented a method to detect the zones of interest in real images where a given queried symbol appear. We have modelled a vectorial signature in terms of accumulated length and angle to guarantee invariance to scale, and we have proposed a similarity measure using a cross correlation to be invariant to rotation. Finally a voting scheme has been used to cluster the zones of interest where different parts of the queried symbol appear. The experimental results show that this approach can detect graphical symbols appearing in low-resolution real images.

The combination of a symbol spotting method, as the presented one, with some well-known shape descriptor yielding good recognition results, makes viable the developing of applications able to detect and recognize graphical symbols in real environments. Driving assistance or blind person aid systems are potential applications with needs to recognize symbols in real images acquired with low-resolution devices.

Acknowledgments

This work has been partially supported by the Spanish project TIN2006-15694-C02-02, the integrated project HF2004-0188 and the French project SymbolRec INRIA équipe associée.

References

[1] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattenrn Recognition*, 13(2):111–122, 1981.

[2] L. J. Latecki, R. Lakämper, and U. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 424–429, 2000.

[3] J. Lewis. Fast template matching. In *Vision Interface*, pages 120–123, 1995.

[4] O. Lorenz and G. Monagan. A retrieval system for graphical documents. In *Symposium on Document Analysis and Information Retrieval*, pages 291–300, 1995.

[5] S. Lucas. Icdar 2005 text location competition results. In *Eighth International Conference on Document Analysis and Recognition (ICDAR)*, pages 80–84. IEEE Computer Society, 2005.

[6] F. Mokhtarian, S. Abbasi, and J. Kittler. Robust and efficient shape indexing through curvature scale space. In *Proc. British Machine Vision Conference*, pages 53–62, 1996.

[7] P. Rosin and G. West. Segmentation of edges into lines and arcs. *Image and Vision Computing*, 7(2):109–114, May 1989.

[8] M. Rusiñol and J. Lladós. Symbol spotting in technical drawings using vectorial signatures. In *Graphics Recognition. Ten Years Review and Future Perspectives*, LNCS 3926, pages 35–46. 2006.

[9] F. Stein and G. Medioni. Structural indexing: Efficient 2-d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(12):1198–1204, 1992.

[10] H. J. Wolfson and I. Rigoutsos. Geometric hashing: An overview. *IEEE Computational Science & Engineering*, 4(4):10–21, October-December 1997.

[11] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37:1–19, 2004.