Word and Symbol Spotting Using Spatial Organization of Local Descriptors

Marçal Rusiñol
Computer Vision Center
Dept. Ciències de la Computació
Edifici O, UAB, 08193 Bellaterra, Spain
marcal@cvc.uab.es

Josep Lladós
Computer Vision Center
Dept. Ciències de la Computació
Edifici O, UAB, 08193 Bellaterra, Spain
josep@cvc.uab.es

Abstract

In this paper we present a method to spot both text and graphical symbols in a collection of images of wiring diagrams. Word spotting and symbol spotting methods tend to use the most discriminative features to describe the objects to be located. This fact makes that one can not tackle with textual and symbolic information at the same time. We propose a spotting architecture able to index both words and symbols, inspired in off-the-shelf object recognition architectures. Keypoints are extracted from a document image and a local descriptor is computed at each of these points of interest. The spatial organization of these descriptors validate the hypothesis to find an object (text or symbol) in a certain location and under a certain pose.

1 Introduction

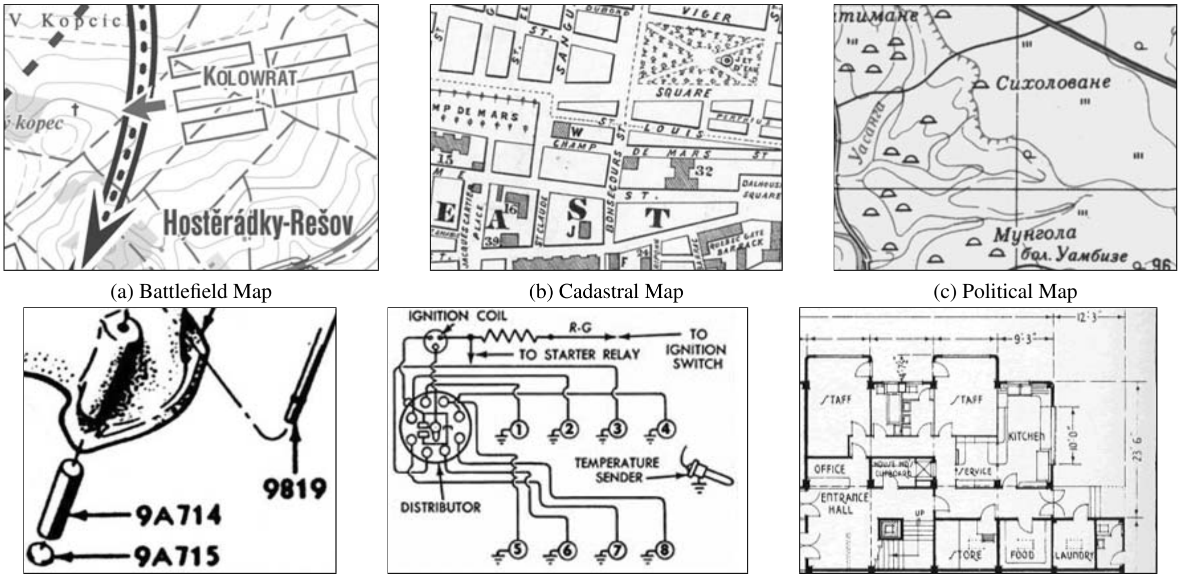
Nowadays, a lot of information is still stored in paper format and great efforts are made to digitalize all these documents. Digitalizing such documents solve problems of space saving and preservation. However, the accessibility to digitalized documents remains an opened issue. The raw image format is not rich enough to allow an easy way to modify or to browse digitalized documents. When talking about images containing mostly textual information, there is the possibility to apply an OCR software. Commercial or open source OCR engines [3] perform good enough recognition rates to obtain a correct text file from a digitalized textual image. Such files can be easily navigated, edited and organized in a database allowing the retrieval of some queried text. However, in the last years, a growing interest is emerging in indexing text documents without performing substring matching processes in the ASCII information resulting from OCR engines, but searching keywords at image level. Such approaches are known as *Word* or *Keyword Spotting*.

But not all the information found in documents has a textual nature. Mechanical and electronic diagrams, architectural floor plans and maps at large contain most of their information in graphical format. Following the idea of word spotting, Tombre and Lamiroy introduced in [24] the concept of *Symbol Spotting*. Symbol spotting techniques are intended to index a large collection of graphical documents in terms of the graphical symbols which appear in it . Excellent recognition rates are not required, the main objective is to have a first coarse retrieval of regions of interest. Given a single instance of a symbol (usually cropped from a document itself) queried by the user, the system has to return a ranked list of segmented locations where the queried symbol is probable to be found.

Both word and symbol spotting are emerging topics in document image analysis field. Each topic uses its own techniques to achieve the localization. Word spotting benefits from the fact that text strings are one-dimensional structures and can apply one-dimensional compact codes as Lu and Tan do in [12] to easily retrieve similar words to the queried one. On the other hand, symbol spotting benefits from the fact that symbols are synthetical entities consisting of uniform regions which are usually highly structured. These facts make geometric relationships between primitives a discriminative cue to spot symbols, as shown by Dosch and Lladós in [4].

However, in Fig. 1 we can see an example of several documents which represents information using both symbolic and textual data. As far as we know, there is no work which try to perform both word and symbol spotting without the need of a previous text/graphics separation [20]. All the works on spotting are based in an ad-hoc description technique depending on the data to tackle with. The use of such strong assumptions provokes that if an architecture has been designed to spot words, it will not detect graphical symbols and viceversa.

The main contribution of this work is to present a spotting architecture which is able to tackle with both words and graphical symbols. Obviously, the use of more general description techniques will affect to the final performance in



(a) Battlefield Map (b) Cadastral Map (c) Political Map

(d) Mechanical Diagram (e) Wiring Diagram (f) Floor Plan

Figure 1: Several documents which represents information using both symbolic and textual data.

comparison to a dedicated method. But on the other hand, the presented system is able for instance to locate text despite the classical similarity transforms one has to face when dealing with symbols.

The main idea of the presented method is to discriminate graphical objects by the spatial arrangement of some classical local descriptors computed over interest points. After the keypoint extraction and the local descriptor computation, a hash function aims to form equivalence classes of similar keypoints. The use of a hashing technique allow us to quickly index and retrieve the location of keypoints by similarity. By querying the local descriptors pairwise, we obtain information about the spatial organization of the descriptors which compound an object. A voting scheme validates the hypothetic locations where to find an object under a certain pose. To conduct the experimental results, we will focus on an document image database arising from wiring diagrams of the automotive industry.

The remainder of this paper is organized as follows: we briefly review some related work in section 2. In section 3 the keypoint extraction and the local descriptors we use are described. Subsequently, in section 4, we present how the local descriptors are organized in an indexing structure to easily retrieve similar interest points. In section 5 we describe how the spatial organization of local descriptors is used to spot objects in the document database. Section 6 provides the experimental results and finally section 7 is a summary and discussion of extensions and future work.

2 Related Work

As mentioned above, several works can be found in both word an symbol spotting. Despite the main idea of spotting words and symbols remain the same, we will see that the techniques used to solve a problem or another are really different. This fact makes that when one wants to tackle with textual and symbolic information, usually the adopted solution is to have a first step to separate text from graphics. The final spotting results are obviously very affected by the performance of this preprocessing step. Lets briefly review some of the spotting techniques we can find in the literature.

2.1 Word Spotting

OCR engines benefit from the nature of alphanumeric information, i.e. text strings are one-dimensional structures with underlying language models that facilitate the construction of dictionaries and indexing structures and so do word spotting techniques. The main idea is to represent keywords with shape signatures in terms of image features. The detection of the keyword in a document image is done by a crosscorrelation approach between the prototype signature and signatures extracted from the target document image.

Although using image features without word segmentation, the information is still one-dimensional and facilitate the use of some classical techniques used in speech recognition. Rath and Manmatha describe in [19] handwritten words by their normalized projection profiles. These word signatures are seen as time series and are aligned using the Dynamic Time Warping (DTW) distance.

Kuo and Agazzi use in [7] another classical technique of the speech processing field. A Hidden Markov Model (HMM) is used to spot words in poorly printed documents. In this case, a learning step to train the HMM is needed. In addition each word the user wants to query has to be learned previously.

Lladós and Sánchez propose in [9] a keyword spotting method based on the shape context descriptor. Words are represented by a signature formulated in terms of the shape context descriptor and are encoded as a bit vectors codewords. A voting strategy is presented to perform the retrieval of the zones of a given document containing a certain keyword.

Recently, Lu and Tan propose in [12] a very simple typewritten word coding which is useful enough to characterize documents. The proposed word code is based on character extremum points and horizontal cuts. Words are represented by simple digit sequences. Several similarity measures based on the frequency of the codes are defined to retrieve documents written in the same language or talking about similar topics.

One of the weak points we find in almost all the methods presented in the existing literature, is that most of the approaches start from a word segmentation (which can be later refined). Very few methods can deal with the document image as a whole. We believe that rather than a crosscorrelation approach, the use of some indexing structure pointing to the locations where the queried word appear would be much more interesting for spotting purposes. As we will see, some symbol spotting methods are based on this idea.

2.2 Symbol Spotting

In the Graphics Recognition community, literature dealing with symbol recognition is vast. Symbol descriptors yielding excellent recognition rates in front of extreme distortions can be found. However usually such descriptors are computationally expensive and only work with isolated data. The main idea of symbol spotting is to describe symbols by a very coarse descriptor to foster the querying speed rather than the recognition rates. Even if symbol spotting is still an emerging topic, several works facing the problem can be found.

Some techniques work with a previous ad-hoc rough segmentation, as in [22, 23], between text and graphics, or thick and thin lines to separate symbols from background. Global numeric shape descriptors are then computed at each location and compared against the training set of pixel features extracted from model symbols. As most of the word spotting methods, in this case, when querying a certain object, a set of segmentations are proposed. A descriptor is computed sequentially for each sub-image and a distance metric decides whether if it is the searched item or not. These techniques lacks of flexibility and won't be a feasible solution to adopt when facing large collections.

Other techniques as in [1, 8, 13] rely on a graph based representation of the document images. These methods focus on a structural definition of the graphical symbols. Subgraph isomorphism techniques are then proposed to locate and recognize graphical symbols with a single step. However these approaches do not seem suitable when facing large collection of data since graph matching schemes are computationally expensive.

Realizing that the computational cost has to be taken into account, several works as [4, 21, 26] are centered on computing symbol signatures in some regions of interest of the document image. These regions of interest can come from a sliding window or be defined in terms of junction points. Obviously, these methods are quicker than graph matching or sequential search, but make the assumption that the symbols always fall into a region of interest. In addition, symbol signatures are usually highly affected by noise or occlusions.

Our feeling, as in the case of word spotting, is that indexing mechanisms and voting schemes are very useful when trying to not only recognize a graphical object but when trying to locate and recognize at the same time.

2.3 Spatial Organization of Feature Points in Document Analysis

It has been shown that the spatial organization of invariants computed from keypoints is a powerful tool to recognize objects in scenes and to index images in terms of their contents. The use of affine invariants allow to tackle with images affected by scale and perspective changes like in [14].

In the document analysis field, Nakai, Kise and Iwamura introduced in [16, 17] a method to retrieve document images acquired with a camera from a large image database using the arrangement of invariants computed over extracted feature points. The results are promising in terms of accuracy, time and scalability.

In this paper, we present a similar approach which aims to index subparts of technical documents and which is able to tackle with both graphical symbols and text.

3 Invariant Keypoints Extraction and Local Descriptors

Most of object recognition methods rely on three basic steps. First interest points are extracted and taken as primitives. At each of these points a local descriptor is computed. The second step performs a matching between descriptors of the object model and descriptors of the image. Finally, a voting scheme validates the hypothetic locations and finds the objects under a certain pose. The spotting method we present is inspired by this quite standard methodology.

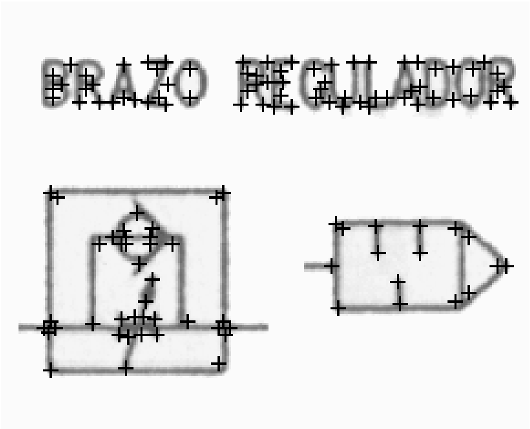


Figure 2: Keypoint extraction using the Harris-Laplace detector over symbols and text. For visibility only the centers of the regions of interest are shown.

As object recognition problem deals with real images, the keypoint extraction must be tolerant to scale and affine transforms to be able to face images affected by perspective changes. On the other hand, the computed descriptors also have to be invariant to similarity and affine transforms. We can find in the literature several approaches to extract interest points in a given image, as for instance in [11, 15]. However, in our case, the symbols and the text we find in technical documents are only affected by changes of scale and rotation transforms since affine deformations can not happen. We use the Harris-Laplace detector described in [15] that extracts points with high curvatures (as corners or junctions) and automatically selects the scale of the region to compute the local descriptor. We can see in Fig. 2 some examples of extracting keypoints in both graphical symbols and text.

The next step is to compute a local descriptor centered in each of the interest points. For our experiments, we tested several off-the-shelf descriptors, namely, the SIFT features [10], shape contexts [2], Hu's geometric moment invariants [6] and a set of steerable filters [5].

Contrary to object recognition methods, we do not match the model descriptors against all the computed image descriptors. As we want to spot graphical objects in a database of document images, the amount of descriptors is huge. A brute force matching is not a feasible solution. The main idea is to use an indexing structure to efficiently retrieve a set of interest points with similar descriptor than the query. The main point is to foster the querying speed rather than having a very accurate description of primitives.

4 Indexing Local Descriptors by Similarity

A grid file [18] structure or a hash table, is used to coarsely quantize the n-dimensional descriptor space, and quickly retrieve all the keypoints having a similar description. To avoid boundary effects when hashing descriptors, each keypoint is stored into the two closest buckets in each dimension. Obviously, the selection of one or other descriptor will affect to the final result, but by means of the quantization we easily simplify the classical descriptors to be able to use them as indexes of a hash table.

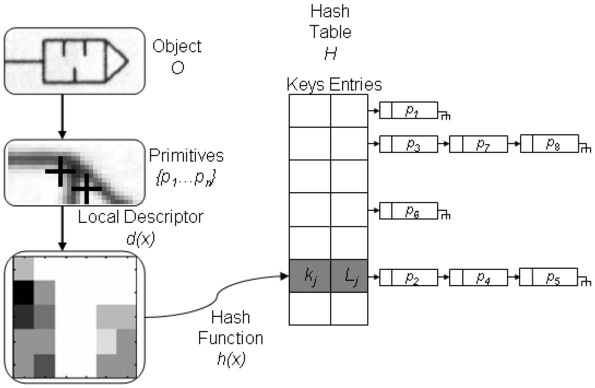


Figure 3: General architecture of our spotting system. The location of the keypoints are stored in the buckets of the indexing structure. Keypoints having similar description share the same table entry and can be easily retrieved by similarity.

Formally speaking, we denote a textual or symbolic object as a set of keypoints $O = \{p_1...p_n\}$. At each keypoint a local descriptor $d(x)$ is computed. A hash function $h(x)$ quantizes the n-dimensional description space projecting each descriptor into an index k_j

$$h(d(p_i)) = k_j,\ i \in [1, n],\ j \in [i, m],\ m \leq n$$

The hash table H uses k_j as indexes and stores at each entry a list

$$L_j = [p_1...p_r]$$

containing all the keypoints having the same index and thus a similar description. We can see in Fig. 3 an overview of the keypoint extraction and local descriptors organization.

The presented spotting system is conceived to query graphical objects in a document image database. The indexing structure is easily extended to manage a third dimension representing diverse document images. When searching an object, all the n keypoints which compound the object are queried in the hash table resulting in a list of locations of several documents where to find similar keypoints.

The use of this kind of indexing structures aims to efficiently retrieve the location of a set of keypoints having a similar description to the queried ones. However this is not enough to locate and segment a certain object. The main idea is to define an object not only by means of the description of a certain points of interest, but by their spatial organization.

5 Spatial Organization of Local Descriptors

To represent the spatial organization of local descriptors, a proximity graph is constructed. Each keypoint is linked to its k nearest keypoints by an edge of the graph. We can see an example of this graph in Fig. 4.

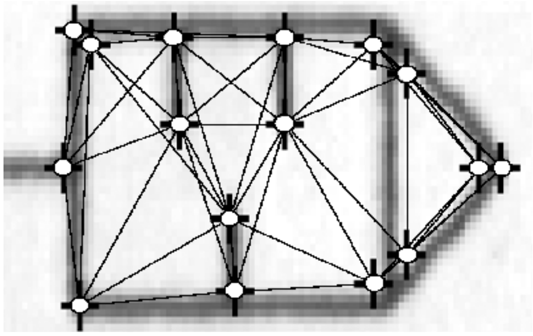


Figure 4: A proximity graph ($k = 5$) is computed from the extracted keypoints. Each keypoint has an associated index in the hash table. Using the graph and the indexes we have information on the spatial organization of similar local descriptors.

Formally speaking, we define the proximity graph $G(O) = (V, E)$ of an object as the structure which represents the spatial organization between close keypoints. A node $n_i \in V$ represents the corresponding keypoint p_i and can be easily retrieved since it can be indexed by the index k_j. An edge $e_{uv} \in E$ between n_u and n_v represents that the keypoints p_u and p_v are close. The edges are attributed by the relative positioning of a keypoint in respect to the other.

The main idea is to use the edges of the proximity graph to perform pairwise queries of keypoints represented by their local descriptors. When querying a tuple of keypoints not only the local information is retrieved (i.e. the keypoints having a similar local description) but also the spatial organization information is used to discriminate a certain object. The presence in the document of two keypoints having similar descriptions than the queried ones and having the same spatial organization accumulate evidences in the hypothetic center of the object to retrieve.

When querying a certain object we follow the idea of the generalized Hough transform. Each pair of keypoints define a basis to compute the hypothetic center hc of the object learned from the model.

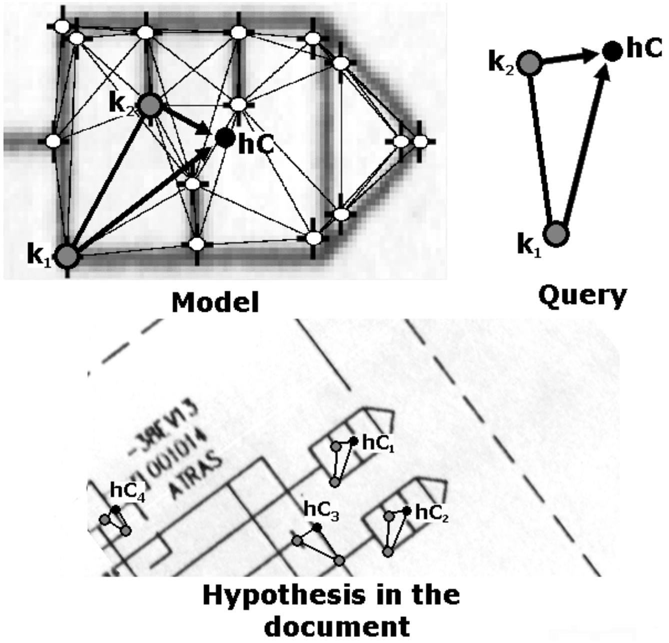


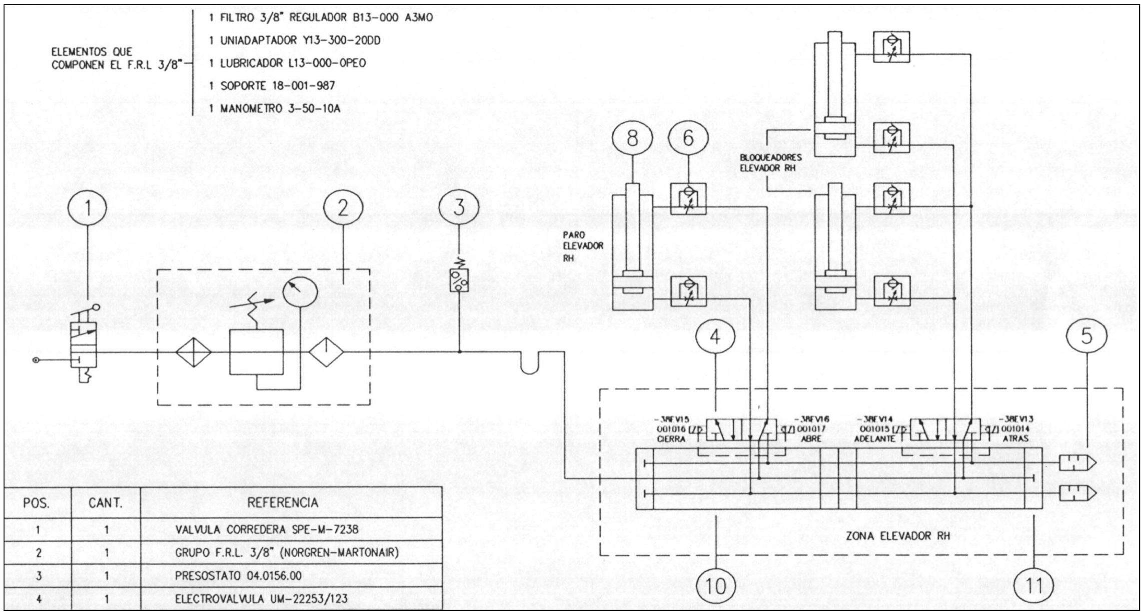
Figure 5: Accumulating evidences of a certain spatial organization of descriptors via pairwise queries.

Given a model to query, each pair of keypoints (k_i, k_j) are queried resulting in a set of hypothetic centers hc_{ij} where to accumulate votes. This voting mechanism accumulates evidences in the locations where we find keypoints having similar descriptions than the queried ones and where the spatial organization of these keypoints can be aligned with the spatial organization of the queried model. The presence of an object under a certain pose in a document provokes a peak in the voting space.

6 Experimental Results

Our experimental framework consisted of two scenarios. First we present some qualitative results which aim to test the performance of the system retrieving words and symbols from a single wiring diagram. Secondly, we present some precision and recall plots which aim to evaluate the performance of the system when querying words and symbols against a collection of documents. In this paper we presented a work in progress, and the tests conducted tests use a small database. We queried six objects (three symbols and three words) in a collection of ten scanned wiring diagrams of 3500×2500 pixels.

In Fig. 6 we can see an example of one of the wiring diagrams used in our experiments. Fig. 7 shows the first ten



POS.	CANT.	REFERENCIA
1	1	VALVULA CORREDERA SPE-M-7238
2	1	GRUPO F.R.L. 3/8" (NORGREN-MARTONAIR)
3	1	PRESOSTATO 04.0156.00
4	1	ELECTROVALVULA UM-22253/123

Figure 6: An example of one of the wiring diagrams used in our experiments.

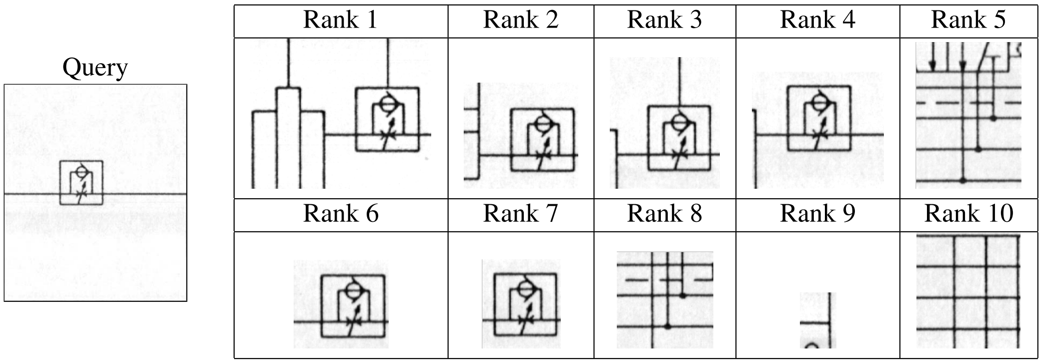


Figure 7: Spotting the flow regulator symbol in the diagram of Fig. 6.

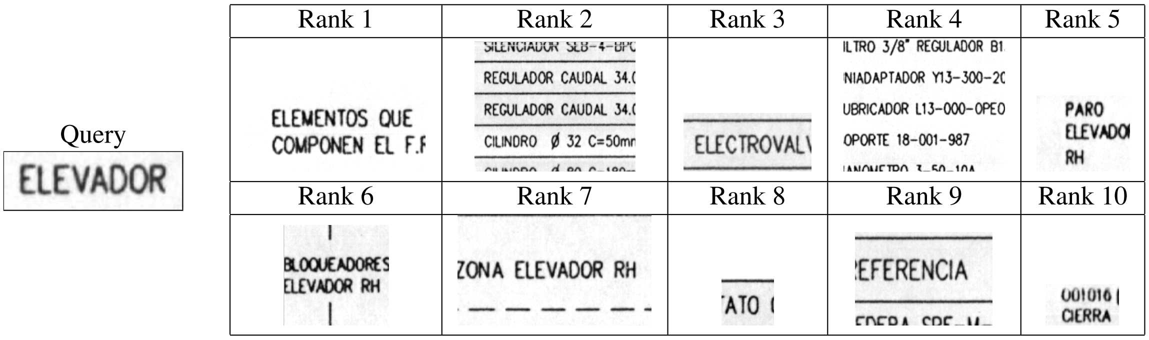
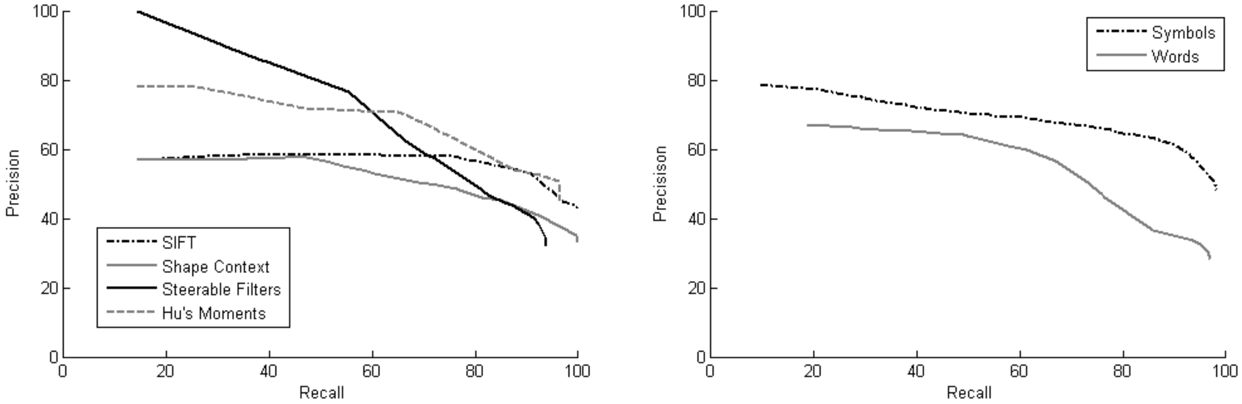


Figure 8: Spotting the word “*elevador*” in the diagram of Fig. 6.



(a) Average precision and recall plots when spotting words and symbols with different local descriptors

(b) Average precision and recall plots when spotting only graphic symbols or word images

Figure 9: Precision and recall plots.

results when querying a flow regulator symbol in the previous diagram. All six instances of the symbol are correctly retrieved, and the false positives which appear are, somehow, understandable since they share some subparts, as the arrowheads, with the queried symbol.

Fig. 8 is more illustrative of the behavior of the presented system. We spot the word "*elevador*" in the example diagram, and we can see that the retrieved locations are all text parts. In addition, we can appreciate that all the results share some elements with the query. For instance, the first location contains the word "*elementos*" which starts just like the query, the second one is "*regulador*" which ends like the query. Besides the correctly retrieved words, other false positive examples are "*electrovalvula*", "*lubricador*", "*adaptador*", "*bloqueadores*", etc.

To have an idea of the performance of the system when querying words and symbols against a collection of documents, we use the common ratios of precision (P) and recall (R) used in the information retrieval field. The *retrieved* items the system results are ranked by the number of votes. Each result image is considered as *relevant* or not depending on its overlapping with the groundtruthed data. Precision and recall are then computed as follows

$$P = \frac{||retrieved \cap relevant||}{||retrieved||}$$

$$R = \frac{||retrieved \cap relevant||}{||relevant||}$$

The precision metric measures the quality of the system in terms of the ability of the system to only include relevant items in the result, whereas the recall ratio measures the effectiveness of the system in retrieving the relevant items. For more details of the computation of these measures, the interested reader is referred to van Rijsbergen's [25] book on information retrieval.

We present in Fig. 9a the average precision and recall plot of querying both words and symbols in the whole database using several local descriptors. Fig. 9b shows the average precision and recall plot for all the local descriptors when querying only words or symbols.

We can appreciate that the performance of the system varies depending on which kind of local descriptor we apply. Obviously the quantization performed by the hash function also can affect the results, but in these experiments we maintain the same sampling criteria.

Simpler descriptors as geometric moments of steerable filters are very sensible to noise which provokes that we only retrieve very similar keypoints. This lack of flexibility provokes that some objects are missed since their keypoints can not be retrieved from the indexing structure. For small recall values, the precision is very high, but it quickly falls, and a 100% recall can not be reached.

On the other hand, more sophisticated descriptors as shape context or SIFT features provide a more stable response with a lower precision. This low precision values indicates that false positives are also retrieved. On the other hand we can also appreciate that the response of the system when querying only graphics or only text is more or less stable. Both curves of Fig. 9b have the same tend, despite the difference of the initial precision. This difference is understandable, since as we seen in Fig. 8 when querying words coherent false positives tend to appear with more frequency that when querying graphics.

7 Conclusions

In this paper we presented a method to spot graphical objects appearing inside the document image database. Basically, we used the spatial organization of off-the-shelf local descriptors as the discriminant features which locates a certain object under a certain pose. The proposed method is flexible enough to aim to spot whether graphical symbols or words. We tested our method in a document image database arising from wiring diagrams of the automotive industry.

Our feeling is that the use of indexing structures such as hash tables and voting schemes has to be one of the mainstays of spotting architectures. In addition, the combination of a coarse local description and geometric information seems to have a high discriminative power. The promising results encourages us to further research following this direction.

Acknowledgments

This work has been partially supported by the Spanish projects TIN2006-15694-C02-02 and CONSOLIDER-INGENIO 2010 (CSD2007-00018).

References

[1] E. Barbu, P. Hérroux, S. Adam, and S. Trupin. Frequent graph discovery: Application to line drawing document images. *Electronic Letters on Computer Vision and Image Analysis*, 5(2):47–57, 2005.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(24):509–522, 2002.

[3] T. Breuel. The ocropus open source ocr system. In *IS&T SPIE 20th Annual Symposium, Document Recognition and Retrieval XV*, pages 0–0, 2008.

[4] P. Dosch and J. Lladós. Vectorial signatures for symbol discrimination. In *Graphics Recognition*, LNCS 3088, pages 154–165. 2004.

[5] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[6] M. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8:179–187, 1962.

[7] S. Kuo and O. Agazzi. Keyword spotting in poorly printed documents using pseudo 2-d hidden markov models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(8):842–848, 1994.

[8] J. Lladós, E. Martí, and J. Villanueva. Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(10):1137–1143, 2001.

[9] J. Lladós and G. Sánchez. Indexing historical documents by word shape signatures. In *Ninth International Conference on Document Analysis and Recognition, ICDAR*, pages 362–366, 2007.

[10] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision, ICCV*, pages 1150–1157, 1999.

[11] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[12] S. Lu and C. Tan. Retrieval of machine-printed latin documents through word shape coding. *Pattern Recognition*, 41(5):1816–1826, 2008.

[13] B. Messmer and H. Bunke. Automatic learning and recognition of graphical symbols in engineering drawings. In *Graphics Recognition Methods and Applications*, LNCS 1072, pages 123–134. 1996.

[14] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Eight IEEE International Conference on Computer Visison, ICCV*, pages 525–531, 2001.

[15] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[16] T. Nakai, K. Kise, and M. Iwamura. Camera-based document image retrieval as voting for partial signatures of projective invariants. In *Eight International Conference on Document Analysis and Recognition, ICDAR*, pages 379–383, 2005.

[17] T. Nakai, K. Kise, and M. Iwamura. Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. In *Document Analysis Systems VII*, LNCS 3872, pages 541–552. 2006.

[18] J. Nievergelt, H. Hinterberger, and K. Sevcik. The grid file: An adaptable, symmetric multikey file structure. *ACM Transactions on Database Systems*, 9(1):38–71, 1984.

[19] T. Rath and R. Manmatha. Word image matching using dynamic time warping. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 521–527, 2003.

[20] P. Roy, J. Lladós, and U. Pal. Text/graphics separation in color maps. In *International Conference on Computing: Theory and Applications, ICCTA*, pages 545–551, 2007.

[21] 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.

[22] S. Tabbone and L. Wendling. Recognition of symbols in grey level line-drawings from an adaptation of the radon transform. In *International Conference on Pattern Recognition, ICPR*, pages 570–573, 2004.

[23] S. Tabbone, L. Wendling, and K. Tombre. Matching of graphical symbols in line-drawing images using angular signature information. *International Journal on Document Analysis and Recognition*, 6(2):115–125, 2003.

[24] K. Tombre and B. Lamiroy. Graphics recognition - from re-engineering to retrieval. In *Seventh International Conference on Document Analysis and Recognition, ICDAR*, pages 148–155, 2003.

[25] C. van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann Newton, MA, USA, 1979.

[26] D. Zuwala and S. Tabbone. A method for symbol spotting in graphical documents. In *Document Analysis Systems VII*, LNCS 3872, pages 518–528. 2006.