A Symbol Spotting Approach Based on the Vector Model and a Visual Vocabulary

Thi Oanh Nguyen, Salvatore Tabbone, Alain Boucher

A Symbol Spotting Approach Based on the Vector Model and a Visual Vocabulary

Thi-Oanh Nguyen[1,2], Salvatore Tabbone[1] and Alain Boucher[2,3]
[1]LORIA - Université Nancy 2 , Campus scientifique - BP 239, 54506 Vandoeuvre-les-Nancy, France.
[2]Institut de la Francophonie pour l'Informatique, MSI, UMI 209 UMMISCO, Hanoi, Vietnam
[3]IRD, UMI 209 UMMISCO, IRD France Nord, Bondy, F-93143, France
{nguyenth,tabbone}@loria.fr, alain.boucher@auf.org

Abstract

This paper addresses the difficult problem of symbol spotting for graphic documents. We propose an approach where each graphic document is indexed as a text document by using the vector model and an inverted file structure. The method relies on a visual vocabulary built from a shape descriptor adapted to the document level and invariant under classical geometric transforms (rotation, scaling and translation). Regions of interest selected with high degree of confidence using a voting strategy are considered as occurrences of a query symbol. Experimental results are promising and show the feasibility of our approach.

1. Introduction

Symbol retrieval in technical documents is still a hot challenge in the document analysis community. Although well-known symbol descriptors work well to describe isolated symbols, their performance in real applications drop away when symbols are embedded in documents. Extending these descriptors to document level cannot be done straightforwardly. Most of them cannot be directly applied without a previous segmentation step.

For this reason, several approaches based on symbol spotting strategies have been proposed during the last few years. A common scheme is to decompose the document into components and then apply a descriptor on each component. Indeed, a vectorization step is needed for most of these strategies. For instance, Wenyin et al. [12] introduced a spotting method including a learning step and an user feedback process. In [11, 8, 9], only symbols which satisfy some conditions (e.g.. convexity, connectivity or closure) are retrieved. A topology graph used in [4] for submatching or coarse specification of query provides good results; in [7], a modification of a shape representation based on Shape Contexts(called SCIP descriptor) is introduced in order to describe a graphic symbol using visual words to *encode* the image content. However, no real evaluation on non-segmented symbols has been really done in these two previous works.

In this paper, we tackle the symbol spotting problem from a point of view where neither symbol hypothesis nor vectorization step is needed. First, to describe local information in document, we propose an extension of SCIP descriptor to document level. In the literature, techniques based on the concept of visual words have been studied in many researches related to video/image retrieval [1, 5, 10]. We exploit also these techniques for indexing graphic documents and for spotting non-segmented symbols into documents. In this part, a new point in our work is that we propose a strategy based on "fuzzy" word matching in order to mitigate the negative effect of clustering step. Finally, we introduce the interest region detection to support the voting system construction in order to locate occurrences of a query symbol.

This paper is organised as follows. In Section 2, an extension of SCIP descriptor for representing local shape information in graphic documents is introduced. Afterwards, we apply a textual approach to describe graphic documents (Section 3). Particularly, we describe the "fuzzy" word matching strategy. The details of symbol spotting system is addressed in Section 4. The next section (Section 5) is dedicated to the evaluations of the approach. Finally, conclusions are stated in Section 6.

2. Local descriptor

First of all processing, we must choose a descriptor well adapted to graphic symbols for representing the document content. The choice of a particular representation scheme is usually driven by the need to cope with requirements such as robustness against noise, stability with respect to small

distortions, invariance to common geometrical transformations or tolerance to occlusions. In [7], the SCIP descriptor is proposed as a descriptor well-suited to the graphic symbols. It contains rich information about the local geometry of symbols. In this section, we propose an extension of SCIP to describe a graphic document. This descriptor is adapted to non-segmented symbols and guarantees invariance to scaling and rotation.

Recall of SCIP at symbol level: In [7], an object (symbol) $\mathcal{O}$ is represented by a set of SCIP calculated at interest points. The interest points are detected by the DoG (Difference-of-Gaussian) detector [6].

$$\mathcal{O} \equiv \{h_i | p_i \in \mathcal{IP}\} \tag{1}$$

where $\mathcal{IP} = \{p_1, p_2, .., p_M\}$ is the set of interest points, and h_i is the SCIP computed at p_i.

$$h_i(l) = \#\{q_j \neq p_i, q_j \in \mathcal{C} : (q_j - p_i) \in bin(l)\}, l = \overline{1, L} \tag{2}$$

$\mathcal{C} = \{q_1, q_2, .., q_n\}$ is the set of object contours points.

SCIP at document level: At object level, the SCIP is invariant to scaling and rotation thanks to the normalisation of shape context by considering all the contour points of the symbol. However, at document level this strategy cannot be applied, since symbols have not been segmented. By construction, shape contexts describe objects nearby a reference point. Contour points far from a reference point provide less useful information to discriminate objects. Hence, it seems natural to define a neighbourhood region for each reference point. But, how to define this region is not straightforward. We must ensure invariance to scaling of the descriptor computed inside the region and therefore, the region can not be fixed a priori. We propose to use the scale on which the interest point is detected, δ_i, to determine the neighbourhood region associated with this point. Thereby, if an interest point p_i is represented by $p_i = (x_i, y_i, \delta_i, \theta_i)$, its neighbourhood region $\mathcal{N}_i$ is considered as a circle centred at p_i with radius $R_i = \beta\delta_i$, where β is a constant and its value is defined empirically. Then, the SCIP descriptor, h_i, computed on the neighbourhood $\mathcal{N}_i$ is the L-bin histogram:

$$h_i(l) = \#\{q_j \neq p_i, q_j \in \mathcal{C} \cap \mathcal{N}_i : (q_j - p_i) \in bin(l)\}, \tag{3}$$

$l = \overline{1, L}$, and therefore, a document is now described by a set of extended SCIP descriptors computed on interest points:

$$\mathcal{D} \equiv \{h_i | p_i \in \mathcal{IP}\} \tag{4}$$

where $\mathcal{IP}$ and $\mathcal{C}$ denote respectively the set of interest points and the set of contour points in the document. Each interest point p_i is located by its coordinates x_i, y_i, the scale δ_i where this point is detected and its dominant orientation θ_i.

3. Document indexing with visual vocabulary

Generally, the number of interest points detected in each graphic document, and hence the number of SCIP descriptors describing each document, may be huge. Besides, since the SCIP descriptor provides a local description of shapes, an exhaustive search on all descriptors will make redundant matching and waste memory and time. In the literature [1, 5, 10], it has been proposed to apply clustering techniques, to group similar descriptors, to define "visual words". This technique permits to avoid redundant matching and save time as well as memory.

3.1. Vocabulary building

First of all, SCIP descriptors are computed, as described in Section 2, for all the documents in the database. Next, similar descriptors are regrouped into clusters. Each cluster is considered as a visual word defined by its centroid. In theory, any clustering technique can be used. We choose K-means for our tests.

3.2. Visual word matching

We would like to match each point with a visual word. However, we are not sure that any clustering method can provide a perfect classification. It depends on the clustering algorithm if it is well-suited to the shape of the clusters. So, since we are in an unsupervised context and in order to minimise the confusion of the clustering, we decide to associate a point with one or more visual words according to the similarity between this point and all the words. It means that a point p_i is assigned to the corresponding words having the same similarities to this point. The confidence degree of each matching depends on the number of matched words and its corresponding similarity as follows:

$$dConf_{p_i,w_j} = \frac{sim_{p_i,w_j}}{\sum_{w_k \in V_i} sim_{p_i,w_k}} \tag{5}$$

where V_i is set of matched words to point p_i and sim_{p_i,w_j} is the similarity between the point p_i and the word w_j.

$$V_i = \{w_j \in \mathcal{V} | \frac{sim_{p_i,w_j}}{sim_{p_i,w_0}} > \epsilon\} \tag{6}$$

with $\mathcal{V}$ is the visual vocabulary, w_0 is the nearest word of p_i and ϵ is a given threshold ($\epsilon = 0.96$ in our experiments).

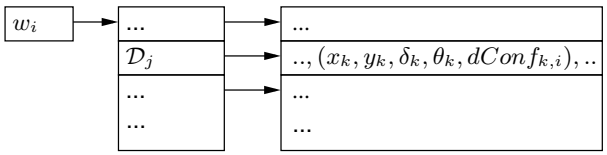


Figure 1. Inverted file structure for a word w_i

3.3. Document indexing

Once the document is described by visual words, it can be treated as a text document. Then, we can apply text retrieval/indexing techniques to our graphic documents. A symbol query is too small in comparison with document. So, in the first step, the vector model is exploited to index the entire document. In the second step, an inverted file structure is used to index the content which helps to match region candidates in each document.

Vector model: In this model, a document is represented as a vector of word frequencies. It is usually described by vector of weighted term frequencies whose component provides the balance of two factors: *term-frequency* (tf factor) and *inverse document frequency term* (idf factor) [2].

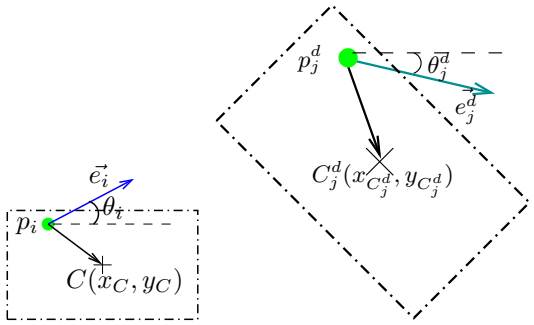
Inverted file: The inverted file structure is composed of two elements: the vocabulary and the occurrences. The vocabulary is the set of visual words built as indicated in the previous section. For each visual word, a list of all occurrence positions where the word appears in documents is stored. Each element of these lists is linked to a document and contains the positions of all the points corresponding to this word with their corresponding degree of confidence. Figure 1 illustrates an entry for a visual word, w_i, in the inverted file.

4. Symbol spotting process

4.1. Regions of interest

Given a query model, the interest points and its corresponding words are determined as described in the previous section. We set the object's centre point $C(x_C, y_C)$ in the query and the corresponding bounding box $rect = (x_C, y_C, w, h)$ (see Figure 2(a)) where w, h denote the width and the height respectively.

Now we try to locate the regions in a document which probably contain the query object. The regions are determined based on the relation between the considered keypoint and $rect$. If point $p_i = (x_i, y_i, \delta_i, \theta_i)$ (in the query) and $p_j^d = (x_j^d, y_j^d, \delta_j^d, \theta_j^d)$ (in the document) are associated with the same visual word (it means that the two points are



(a) query (b) a region in the document

Figure 2. Location of bounding box corresponding to the query in the document.

matched), a region $rect_j^d = (x_{C_j^d}, y_{C_j^d}, w_j^d, h_j^d, \varphi_j^d)$ in the document is determined upon $\{rect, p_i, p_j^d\}$ (see Figure 2), as in (7), via some rotation and translation operations. p_i, p_j^d are considered as control points of the two rectangles $rect$, $rect_j^d$ respectively.

$$\begin{aligned} \varphi_j^d &= \theta_j^d - \theta_i, \;\; w_j^d = w * \xi, \;\; h_j^d = h * \xi \\ x_{C_j^d} &= x_j^d + \xi * (x_{piC} * \cos\varphi_j^d - y_{piC} * \sin\varphi_j^d) \\ y_{C_j^d} &= y_j^d + \xi * (x_{piC} * \sin\varphi_j^d + y_{piC} * \cos\varphi_j^d) \end{aligned} \quad (7)$$

where

$$\begin{aligned} \xi &= \delta_j^d / \delta_i \\ (x_{piC}, y_{piC}) &= (x_C, y_C) - (x_i, y_i) \end{aligned}$$

4.2. Voting process

Based on the vector model [2], the centre of each region of interest is voted on the similarity between this region and the query.

Suppose, that $W_r = \{w_{r_1}, w_{r_2}, .., w_{r_K}\}$ is the set of words matched to the points in a region of interest r with the corresponding degrees of confidence $\{dConf_{r_1}, dConf_{r_2}, .., dConf_{r_K}\}$. The appearance frequency of word w_j in this region is defined by tf_j^r:

$$tf_j^r = \frac{\sum_{w_{r_k} \in W_r, w_{r_k} \equiv w_j} dConf_{r_k}}{\sum_{k=1}^{K} dConf_{r_k}} \quad (8)$$

and the *weighted frequency* of word w_j in this region:

$$wf_j^r = tf_j^r * idf_j \quad (9)$$

where idf_j is the *inverse document frequency* term of word w_j. The region is represented by a vector of weighted frequency v^r.

$$v^r = (wf_1^r, wf_2^r, .., wf_{\|\mathcal{V}\|}^r) \quad (10)$$

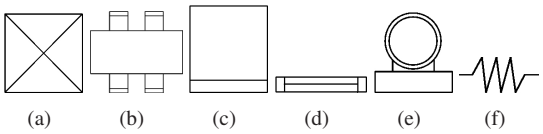


Figure 3. Queries

The cosine distance between v^r and v^q (the vector of weighted frequency for the query object) denotes the vote value for the centre of region r. A vote map is built and and the interest regions corresponding to maximum peaks of this map indicate in the document the occurrences of a query object.

Filtering keypoints As a document is much larger than a symbol, false keypoints can belong to a region because they are detected at very large scale. In order to get representative keypoints of a region, we choose only those for which scales are in the range corresponding to the scale range of the query. If s_{min} and s_{max} are respectively the minimal and maximal scales of a query, for a region ($rect_j^d$ in Figure 2) only the keypoints whose scales are in $[\delta_j^d - (\delta_i - s_{min}), \delta_j^d + (s_{max} - \delta_i)]$ are selected to represent the region ($rect_j^d$ in this case).

5. Experimental results

We present in this section our preliminary experimental results to verify the performance of the spotting system. Our tests are executed on a collection of synthetic documents (see Figure 4) of SESYD project [3].

In Table 1, we show our results obtained with the queries of Figure 3 on fifteen graphic images. The first column indicates the query symbols. nGT is the number of symbol occurrences for each query existing in the documents (the ground truth). nCD and nFP are respectively the number of occurrences detected correctly and incorrectly (false positives). And the fifth column ($nFN = nGT - nCD$) indicates the number of occurrences existing in the document which are not detected (false negatives). We computed also the precision: $P = nCD/(nCD + nFP)$ and the recall: $R = nCD/nGT$ for each query. The precision rate is related to the number of false positives that is the precision increases when the false positives decrease. Similarly, the recall is related to false negatives, the fewer false negatives, the higher the recall is. For computing these values, all detected regions which cover completely or almost completely the correct symbol are considered as correct detections, the others are considered as false potitives. Figure 5 shows some examples of correct and incorrect detections. We can see that the system responds very well to queries

Q.	nGT	nCD	nFP	nFN	P	R
Fig. 3(a)	58	58	0	0	58/58	58/58
Fig. 3(b)	15	15	30	0	15/45	15/15
Fig. 3(c)	56	32	32	24	32/64	24/56
Fig. 3(d)	15	15	30	0	15/45	15/15
Fig. 3(e)	15	15	0	0	15/15	15/15
Fig. 3(f)	38	28	10	10	28/38	28/38
Average	Precision = 0.7, Recall = 0.88					

Table 1. Spotting results for the queries in Figure 3: Q.: Queries, nGT: number of symbol occurrences in the document, nCD: number of correct detections, nFP: number of false positives, nFN: number of false negatives, P: precision, R : recall.

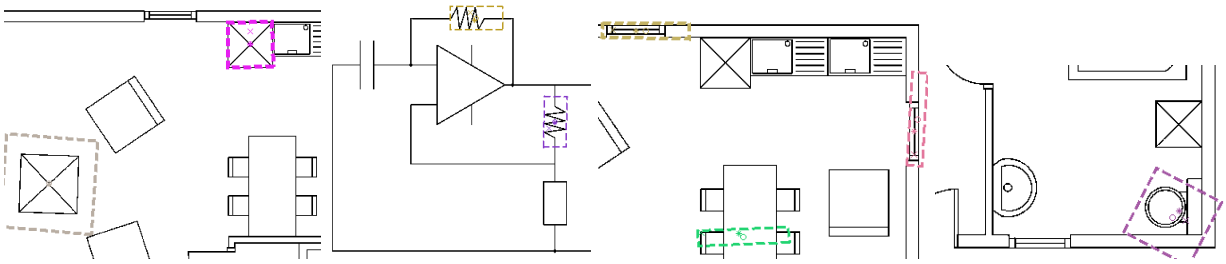
in Figure 3(a), 3(e) and 3(f). For other queries, the precision is not always high but the recall is very good. Both the global recall (88%) and the global precision (70%) are high. It means that our approach can capture almost all occurrences in the document but there are also some incorrect detections. Some visual spotting results are showed in Figure 4.

These incorrect detections are caused by two principal reasons. The first one is that the spatial relations between visual words in a region is not really considered. So, regions containing similar sub-parts with the query are returned as instances of the query. It causes a low precision in the case of query Fig. 3(b), 3(c) (see Table 1, Figure 5(d)) and incorrect detection in Figure 4(c) (the region having a spring-green boundary). Nevertheless, this type of error can be considerably reduced if we take into account the global information of the region or more information about the spatial relations between words in a region. The second reason comes from the instability of the keypoints detection in the case of symbols composed of curves because the DoG detector is appropriate to detect lines saliency but not smoothed configuration. It makes the incorrect orientation of symbol though the symbol is correct (see Figure 4(d)).

However, our approach does not need any hypothesis (as convexity, connectivity,..) on the detected symbols nor vectorization step.

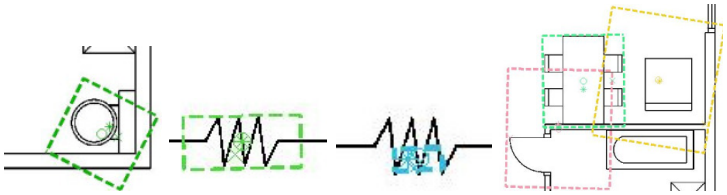
6. Conclusions

We present in this article, a local descriptor to describe graphic documents and propose an approach for symbol spotting. The local descriptor satisfies the affine transformation (rotation, scaling, translation) and provides a good representation of local structures. The spotting system is built by voting on regions of interest and based on the vec-



(a) using query 3(a) (b) using query 3(f) (c) using query 3(d) (d) using query 3(e)

Figure 4. Examples of symbol spotting results.



(a) (b) (c) (d)

Figure 5. Examples of detections considered as correct and incorrect: (a) correct detection for query 3(e); (b) correct detection for query 3(f); (c) incorrect detection for query 3(f); (d) one correct detection (green boundary) and two incorrect detections (golden-rod and light-rose ones) for query 3(b).

tor model with the technique of visual vocabulary. Our approach does not require any constraints for spotted symbol and vectorization step. The results are very promising even if false detections are also provided. So, for our future works, we are interested in integrating the spatial relation to decrease the false positives responses. Also, we want to reduce the instability of keypoints detection when symbols are described by curves in order to recover the missing detections.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, November 2004.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, New York, 1999.

[3] M. Delalandre, T. Pridmore, E. Valveny, H. Locteau, and E. Trupin. Building synthetic graphical documents for performance evaluation revised. In *Selected Papers of Workshop on Graphics Recognition (GREC), Lecture Notes in Computer Science*, volume 5046, pages 288–298. Springer Berlin / Heidelberg, 2008.

[4] J. Fonseca, A. Ferreira, and J. Joaquim. Content-based retrieval of technical drawings. *International Journal of Computer Applications in Technology*, 23(2-3):86–100, March 2005.

[5] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *International Conference on Computer Vision*, volume 1, pages 604–610, October 2005.

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

[7] T.-O. Nguyen, S. Tabbone, and O. T. Ramos. Symbol descriptor based on shape context and vector model of information retrieval. In *DAS 2008*, September 2008.

[8] M. Rusinol and J. Llados. Symbol spotting in technical drawings using vectorial signatures. In *Graphics Recognition. Ten Years Review and Future Perspectives*, volume 3926/2006, pages 35–46. Springer Berlin / Heidelberg, October 2006.

[9] M. Rusinol and J. Llados. A region-based hashing approach for symbol spotting in technical documents. In *Seventh IAPR International Workshop on Graphics Recognition*, September 2007.

[10] J. Sivic and A. Zisserman. Video google: Efficient visual search of videos. In *Toward Category-Level Object Recognition*, volume 4170/2006, pages 127–144. Springer Berlin / Heidelberg, 2006.

[11] S. Tabbone and D. Zuwala. An indexing method for graphical documents. In *International Conference on Document Analysis and Recognition*, volume 2, pages 789–793, 2007.

[12] L. Wenyin, W. Zhang, and L. Yan. An interactive example-driven approach to graphics recognition in engineering drawings. *International Journal of Document Analysis and Recognition*, 9(1):13–29, March 2007.