

Towards Searchable Line Drawings, a Content-Based Symbol Retrieval Approach with Variable Query Complexity

Nibal Nayef^(✉), Wonmin Byeon, and Thomas M. Breuel

Technical University Kaiserslautern, Kaiserslautern, Germany
{nnayef,tmb}@iupr.com,
byeon@rhrk.uni-kl.de

Abstract. Current symbol spotting and retrieval methods are not yet able to achieve the goal of both high accuracy and efficiency on large databases of line drawings. This paper presents an approach for focused symbol retrieval as step towards achieving such a goal by using concepts from image retrieval. During the off-line learning phase of the proposed approach, regions of interest are extracted from the drawings based on feature grouping. The regions are then described using an off-the-shelf descriptor. The similar descriptors are clustered, and finally a visual symbol vocabulary is learned by an SVM classifier. The vocabulary is constructed assuming no knowledge of the contents of the drawings. During on-line retrieval, the classifier recognizes the descriptors of query regions. A query can be a partial or a complete symbol, can contain contextual noise around a symbol or more than one symbol. Experimental results are presented for a database of architectural floor plans.

Keywords: Symbol retrieval · Multi-part graphical query · Visual symbol vocabulary · Technical line drawings

1 Introduction

Symbol recognition and retrieval in technical line drawings is one of the main challenges in document analysis and graphics recognition communities. State-of-the-art methods work well on small databases of isolated symbols, but their performance significantly decreases on large databases of complicated drawings containing many symbols connected to each other and to connection lines. To achieve realistic large scale symbol retrieval, we could use ideas from the field of content-based retrieval in natural image databases. The famous bag-of-visual-words (BoVW) approach [2], with its many variants, have been successfully used in large scale object retrieval and image analysis tasks. The main idea there is not to search images one by one looking for a queried object, but rather to perform off-line content analysis of an image database, and create a compact and quickly accessible representation of the whole database. Such representation

can be searched quickly for a query. The symbol retrieval approach presented in this paper, loosely follows some of the concepts of the BoVW approach. Our approach consists of an off-line and an on-line stage. Within the off-line stage, local regions of interest (ROIs) are extracted from the line drawings, and then described by a shape descriptor. The descriptors are clustered, and then prepared to be quickly accessible based on queries. The idea behind this stage is to create a compact and quickly accessible representation of the contents of the line drawings database. Within the on-line stage, contents similar to images of query symbols are retrieved. Our approach is for focused retrieval, which means retrieval with localization of the queried symbol inside the retrieved line drawings.

The rest of this paper is organized as follows. Section 2 reviews the related work in comparison to the proposed approach. Section 3 explains the overall proposed approach with a detailed description of each step. In the section after that, we present the performance evaluation of the proposed retrieval system. Finally, we present conclusions and directions for future work.

2 Related Work

Some state-of-the-art methods/systems that perform symbol spotting or retrieval, follow the BoVW approach as well. Nguyen et al. [18] proposed a symbol spotting method that is very similar to the BoVW model. In their work, DOG-SIFT [13] is used to detect ROIs. The ROIs are described using shape context descriptor, and then clustered by k-means to build a visual vocabulary. An inverted file index is built from the clusters (similar to the inverted index of text retrieval systems). The location information of the ROIs are also kept in the index, since the spotting system should locate regions in the drawing rather than returning the complete drawings. It is not clear how their method would scale using a larger number of symbols and line drawings.

Kong et al. [10] presented a symbol spotting/retrieval method where the ROIs are found using overlapping sliding windows. The ROIs are then described by a shape descriptor called the deformed blurred shape model. The descriptions are computed at different scales and orientations. Then the vectors describing the ROIs are clustered using k-means. In their method (as well as in the method of Nguyen et al. [18]), k-means is used as a clustering method. The use of such a simple clustering method means relying more on the goodness of the output of the preceding ROI detection and description steps. In [10], those steps use overlapping sliding windows, which are not invariant to scale and rotation. This results in different ROIs that should correspond to the same symbol part. Moreover, the used ROI-descriptor is not invariant to rotation and scale, so, the ROIs have to be stored many times at different scales and orientations. The increased number of ROIs makes it harder for a clustering algorithm to output the correct partitioning of the data. For indexing, hash tables are built from the clusters for each image, where the labels of the clusters are the keys to a hash table. In their system, such a table is used for every image. This requires processing each image of the database sequentially during spotting/retrieval. Moreover, scalability is an issue when using a hash table.

Luqman et al. [14] extracted ROIs from graph-based representations. The ROIs are converted to fuzzy structural signatures. This descriptor is also introduced by them where a structural graph representation is converted to a numerical vector representation. The similar signatures are then clustered together using agglomerative (or hierarchical) clustering method. The city distance is used as a similarity measure of clustering. Labels are assigned to clusters. An inverted file index is used for indexing the contents of the drawings. The keys of this index are learned by a Bayesian network, and the index naturally contains the ROIs locations and the labels of the clusters to which they belong. During the on-line querying, the ROIs of a query are extracted and described, and then classified as a member of some cluster. The corresponding ROIs in the drawings are retrieved by looking up the index.

The retrieval approach presented in this paper, includes detection of regions of interest (ROIs) using feature grouping proposed by the authors [16]. This ROI detection method has been shown to be highly accurate in finding regions that correspond to similar parts of symbols despite transformations. In the description step, we use a pixel-based compact symbol descriptor proposed by Yang [21], that is invariant to transformations and robust to noise. As for creating the compact and quickly accessible representation of the line drawings, we use the concept of the visual symbol vocabulary from the BoVW approach. We create a visual vocabulary of the database by k-means clustering and SVM learning. The presented approach has the ability to deal with different query complexity thanks to the region detection step. A query could contain contextual noise (additional line segments), or could have multiple parts. Retrieval of a query is performed on the each of the query regions.

The authors have also recently presented another symbol retrieval system [17] that has the same structure as the one proposed here. The system in [17] outperforms state-of-the-art methods including the one presented in this paper. However, our work in [17] requires big memory and time resources for the off-line stage, as the clustering step is based on accurate geometric matching of shapes. Additionally, the time required for on-line query retrieval depends on the size of the visual vocabulary, and it also uses geometric matching for comparing a query with the elements of the visual vocabulary. So, in this paper, we investigate techniques for faster query symbol look up time, and we will seek to improve higher localization and retrieval accuracy. The performance of the approach presented here compares favorably to other state-of-the-art methods.

3 Description of the Retrieval Approach

The retrieval approach proposed in this paper has two stages. The steps of the first stage – the off-line stage – are as follows. First, ROIs of line drawings are identified (Subsect. 3.1). The identified ROIs correspond to symbols’ parts up to complete symbols. This step is accomplished via grouping line segment features of the line drawings. The method used for feature grouping was introduced by the authors in [16]. Second, an off-the-shelf symbol descriptor is applied on the

extracted ROIs to represent them in vector format. We use the symbol descriptor introduced by Yang [21]. The descriptors are the visual words of the line drawings database. The next step is applying k-means clustering to group the similar descriptors together. Finally, an SVM classifier is trained on clusters to learn the class labels of the clusters, where each cluster should contain similar visual words corresponding to similar symbol parts.

In the on-line stage, a query symbol is given as input. The ROIs of the query are detected and described using the same methods used in the off-line stage. The trained SVM classifier quickly classifies the descriptors of the query regions as belonging to which clusters. All the similar regions from the drawings database are then retrieved from the clusters. The following subsections describe each step of the retrieval approach in more detail.

3.1 Detection of Regions of Interest

We argue that the ROI-finding step is the most important step within a content-based retrieval system. The ROIs should be detected consistently and accurately such that a region that corresponds to a symbol is always found in different drawings regardless of the position and the transformation of that symbol. If a method is capable of finding ROIs in such a consistent way, then the later steps of describing, clustering and retrieving those regions become easier to develop and would yield better performance.

We will use a region detection method from our previous works in [15,16], which is based on feature grouping. We have illustrated the accuracy of this method and its invariance to transformed regions in [15,16]. In the following we describe how the method works.

Images of line drawings are given as input to the region detection method. First, to vectorize the line drawings, simple preprocessing is applied: edge detection followed by sampling line segments along the edges. The edges may be straight or curved, the process of sampling line segments along the edges converts all the edges to straight line segments. For example, a curved edge will be converted into a set of small straight segments. The line segments are the features of the line drawings. A set of line segments are grouped together if they form a convex shape. Grouping line segments based on convexity was introduced by Jacobs [7,8] to find meaningful shapes or parts of objects. Such feature grouping method is invariant to scaling and rotation. After finding initial groups of line segments that satisfy convexity property, the following cleaning up steps are carried out: removing all groups that have less than 3 segments, and groups that are cyclic repetitions of each other. As a last step, a group whose bounding box falls completely within a bounding box of another group, is discarded.

After these steps, we get a list of groups of line segments. Each group consists of line segment features that were grouped based on the method described above. The goal is to identify ROIs in a line drawing, hence, we define an **ROI** as **the region (bounding box) that contains a group of line segment features**.

Figure 1 shows the output of the grouping method applied on a line drawing. The shaded parts are the ROIs found by the method. In the figure, red

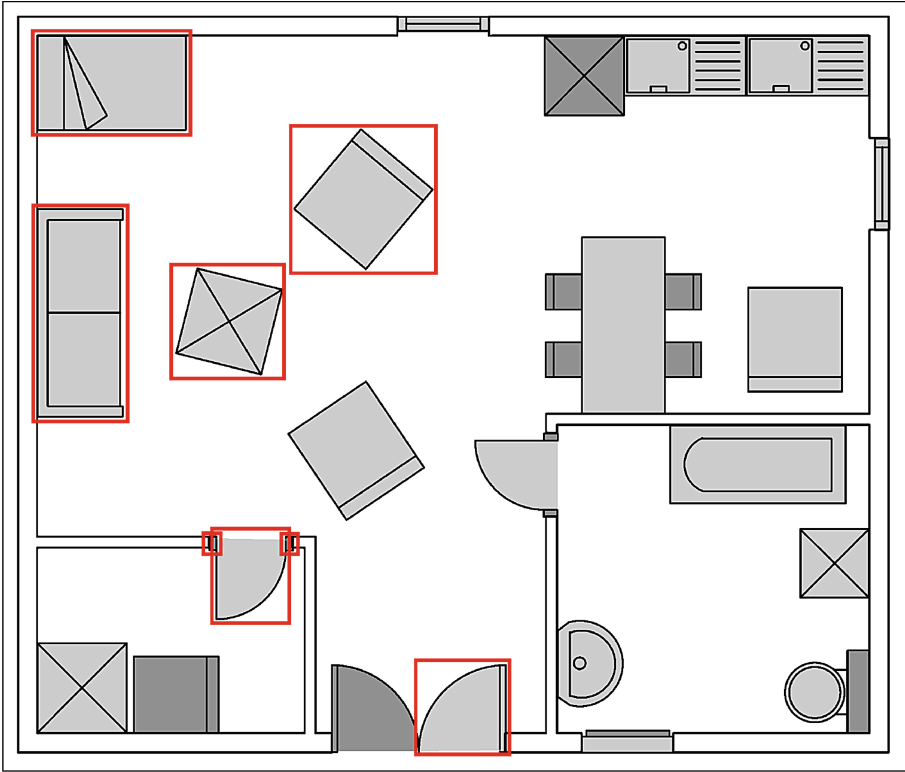


Fig. 1. Output of the region detection method shown on a complete line drawing. The found ROIs (symbols' parts) of a line drawing are shown shaded (adjacent different ROIs have different shading). Red bounding boxes are drawn around **only some** of the found ROIs for clarity of illustration (Color figure online).

bounding boxes are drawn around some of the found ROIs, we also use different shading for adjacent ROIs. It can be seen that the found ROIs correspond mostly to meaningful parts of the symbols up to complete symbols. Note that those symbols appear in different sizes and rotations within a line drawing. For each region, we need to keep information about where this region can be found in a line drawing, i.e. which image it comes from and its location in that image.

3.2 Description of Regions of Interest

The extracted interest regions contain symbol parts up to complete symbols. A shape descriptor is needed to represent a region in a vector form. The vector representation can then be used for the following steps of clustering and learning. Ideally, a symbol descriptor should be used to compute such a vector representation. In this step of our proposed approach, any off-the-shelf symbol descriptor can be used to represent the extracted regions. We have chosen the one

proposed by [21], as it has high accuracy in recognizing symbols. This descriptor is invariant to similarity transformation and is robust to noise as shown in Yang's experiments. In the following we present a brief review of the used descriptor, the complete details can be found in [21].

Yang's symbol descriptor is represented by two histograms. Each histogram contains both statistical and structural information of a shape. Each pixel with its surrounding pixels describes the shape by certain constrains (length-ratio and angle) which provide the detailed structure of the symbol with rotation-scale invariance. In addition, it is robust to noise or distortion by quantization and integration to the fixed dimensions.

The procedure of the descriptor is as follows: (1) shape skeletonization (2) computing pixel-level constraints (3) generating length-ratio (LRH) and angle (AH) based histograms for each pixel (4) statistical integration of all pixels based on LRH and AH. From this integration, we finally obtain two fixed dimensional vectors.

The skeleton of the input shape is sampled into N pixels. Those are N reference points ($P_k, k \in \{0, ..N - 1\}$). Each reference point describes the relationship with all other pixels by length-ratio and angle:

$\{(L_{ij}, \Theta_{ij}) \mid i \in [1, N - 2]; j \in [i + 1, N - 1]\}$ (except P_k), where:

$$L_{ij} = \min \{ | P_k P_i | / | P_k P_j |, | P_k P_j | / | P_k P_i | \} \quad (1)$$

$$\Theta_{ij} = \angle P_i P_k P_j \quad (2)$$

$| P_k P_i |$ and $| P_k P_j |$ are the length of the two vector $P_k P_i$ and $P_k P_j$. The advantage of this representation is that it is not affected by scaling and rotation.

These constraints between every pair of pixels are in a range $[0, 1]$ for length-ratio and $[0, 180]$ for angle. We then construct the matrix to show the distribution of the constraints as well as to generalize the form:

$$LRH = \begin{bmatrix} n_1(P_0) & n_2(P_0) & \cdots & n_M(P_0) \\ \vdots & \vdots & & \vdots \\ n_1(P_k) & n_2(P_k) & \cdots & n_M(P_k) \\ \vdots & \vdots & & \vdots \\ n_1(P_{N-1}) & n_2(P_{N-1}) & \cdots & n_M(P_{N-1}) \end{bmatrix} \quad (3)$$

where

$$n_{\{1..M\}}(P_k) = \frac{2}{(N-1)(N-2)} \text{Hist}_{\{1..M\}}(L_k) \quad (4)$$

$\text{Hist}_{\{1..M\}}(L_k)$ is the histogram for the reference point P_k divided by the number of bins M .

Finally, statistical integration of points $P_{k \in \{0..N-1\}}$ is performed from the previous matrix LRH (SIHA-LRH) and we obtain:

$$SIHA - LRH = \begin{bmatrix} m_1(1) & m_2(1) & \cdots & m_M(1) \\ m_1(2) & m_2(2) & \cdots & m_M(2) \\ \vdots & \vdots & & \vdots \\ m_1(Q) & m_2(Q) & \cdots & m_M(Q) \end{bmatrix} \quad (5)$$

where

$$m_i(1..Q) = \frac{1}{N} \text{Hist}_{\{1..Q\}}(n_i(P_{\{0..N-1\}})) \quad (6)$$

$\text{Hist}_{\{1..Q\}}(n_i(P_{\{0..N-1\}}))$ is the histogram divided as Q number of bins for each column $i \in \{1..M\}$ of LRH. Note that this is applied on the matrix of dimensions $Q \times M$. So, from Eq. 6, we obtain a vector for length-ratios, which is the first part of the descriptor. The vector of angles is constructed in the same manner (SIHA-AH), but the dimension is not necessarily same as SIHA-LRH.

In order to get the final vector form, the two vectors of length-ratio and angle are concatenated into one vector. The same symbol descriptor is applied on all the regions extracted in the previous step. In his work, Yang [21] proposed the sum of absolute differences (SAD) to compare the similarity/difference between descriptors. As will be seen in the next subsection, we use the Euclidean distance (square root of the sum of squared differences) to compare descriptors in the next clustering step. Either distance can be used.

3.3 Clustering and Learning the Classes of the Clusters

At this step of off-line content analysis of line drawings, we have thousands of descriptors of regions of interest. Analogous to the BoVW approach, those descriptors are the visual words of the line drawings. Performing the search in a large space of visual words is computationally expensive, and does not allow for interactive browsing and searching the contents of line drawings. That's why a more compact representation is required in order to provide acceptable search response time. The numerous methods that follow the BoVW approach use techniques such as k-means clustering [12], self-organizing maps [6] and vector quantization [5, 11] for creating compact representations from high dimensional feature spaces [9, 19].

Similarly, we use standard k-means clustering for grouping the similar descriptors together. The input to the clustering method is the list of descriptors – in vector form – computed in the previous step. Recall that each descriptor represents a region of interest, that in turn represents a symbol part up to a complete symbol. Figure 2 shows examples of the constructed clusters.

Clustering parameters – mainly the number of clusters – should be chosen to be proportional to the expected number of regions that could be extracted from the symbol alphabet that is used to draw technical line drawings of some domain. The number of clusters is one of the factors affecting the relation between recall and precision values in the retrieval process. Other enhanced clustering techniques can be used to improve this step. Our goal here is just to develop a framework for retrieval, and show that invariant region detection and description are more critical to the performance of the overall retrieval system.

Once the clusters are created, the only remaining step is to prepare them for query retrieval, such that, when a query descriptor is given, the retrieval system can decide quickly, to which cluster the query belongs. For this purpose, an SVM classifier is used. The classifier is trained on the clusters, where the members of a cluster are the training samples of a class. The class is assigned the same label

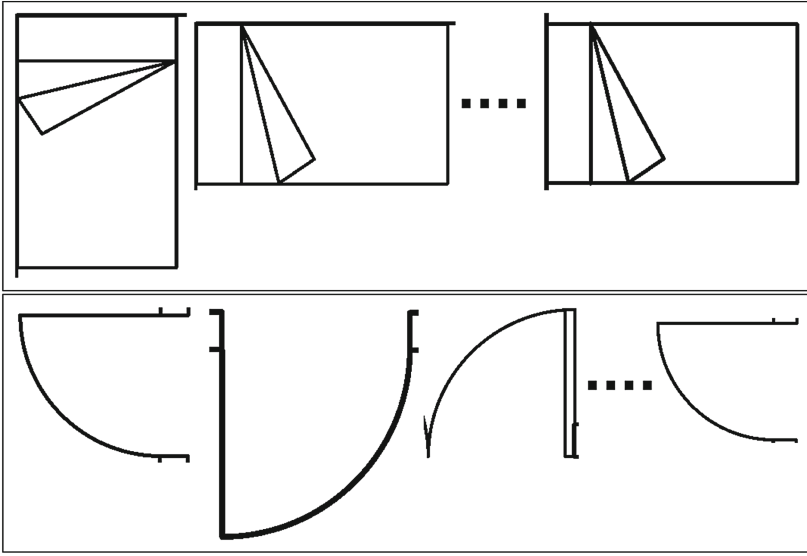


Fig. 2. Examples of the resulting clusters of ROIs (symbols parts). K-means clustering is applied on the descriptors of the ROIs to get such clusters.

of the cluster. The used SVM classifier is from the python MLPY package that implements libSVM [1], with a “linear” kernel type.

Alternatively, the means of the clusters can be used for comparisons with a query. However, it would be slower in case of a large number of classes. Moreover, a learning step in the retrieval framework could be useful in other datasets that might have labeled training data.

3.4 On-line Query Retrieval

A query symbol can be a segmented symbol from the symbol alphabet of the technical drawings, or can be a cropped region from a line drawing. A query that is cropped from a drawing may contain contextual noise around the symbol, such as connection lines or parts from neighboring symbols. The first step to deal with a query, is to extract its regions of interest using the same method discussed in Subject. 3.1. The contextual noise around a query will not be parts of the interest regions. After that, a descriptor is computed from each region of a query as explained in Subject. 3.2. Figure 3 shows the extracted regions of interest for different queries. Extracting the regions and computing the descriptors of the query do not consume much time as a query is usually very small compared to a complete line drawing.

Finally, the regions are classified by the trained SVM classifier. The similar regions from all the drawings are retrieved from the cluster that has the same label. If a query symbol consists only of one region, then no further steps are required. For multi-region queries, a voting procedure or a spatial verification

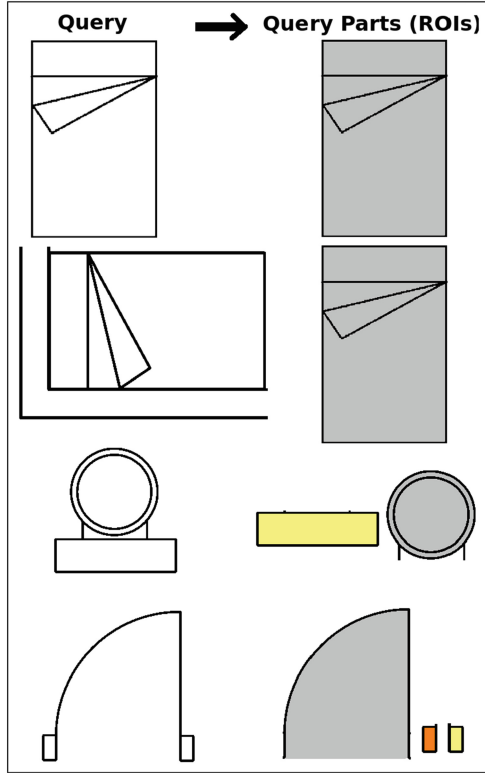


Fig. 3. The left column shows examples of the used query symbols. The second symbol in that column is a cropped symbol from a line drawing, the rest are segmented symbols from the alphabet. The right column shows the extracted regions of interest (ROIs) from the queries on the left.

step can be applied to retrieved regions from different clusters. We have presented a spatial verification step in our recent work in [17] within a geometric-based symbol retrieval system. Such steps can be added to the approach presented in this paper to improve the precision.

4 Performance Evaluation

The database we use for evaluation has 300 images of architectural floor plans. The images are a subset of the graphics recognition database called “Sesyd”. The database is publicly available¹, and it contains different types of technical line drawings. The ground truth is also provided with the database. The ground truth contains information (label and bounding box location) of each symbol in all the drawings.

¹ <http://mathieu.delalandre.free.fr/projects/sesyd/index.html>

The generation of the database is described in [3,4]. The images of the database are synthetic clean images of line drawings that imitate real complete floor plans with sizes between 2M to 7M pixels. Different subsets of this database have been used for evaluation of symbol spotting systems in [14,18]. The GREC-2011 symbol spotting contest² [20] used a set of images very similar to the images of the Sesyd database, as the images of the contest were generated using the same system of [3,4].

The recall and precision measures are used to evaluate the retrieval results of the proposed approach as follows. If the bounding box of a retrieved region contains at least 75% of an instance of the query symbol (or a query region), this is counted as a true positive, otherwise, it is a false positive. Figure 4 shows the recall-precision curve of the retrieval results. Each value on the curve is the average of the recall-precision values obtained for different queries that have a total of 2249 instances in the database of the 300 line drawings.

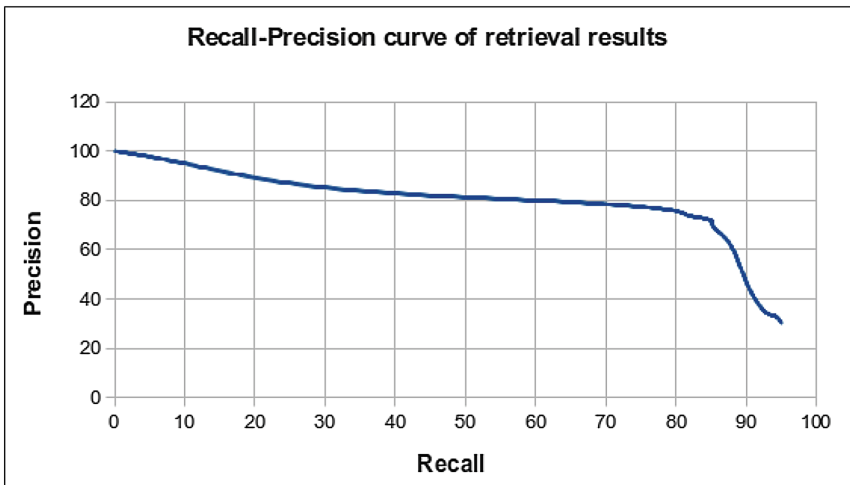


Fig. 4. Retrieval results of the proposed approach. A number of queries of different complexity levels are used, the queries have a total of 2249 instances in 300 line drawings. The shown curve is the average achieved performance of the queries.

The results in Fig. 4 show a range of good performance in terms of both recall and precision. We argue that this is due to the accuracy and invariance of the used region detection method, and also the shape descriptor. We have used simple standard clustering and learning steps (k-means clustering and SVM learning) that were used by other systems for the same purpose, this suggests that the main factors that control the retrieval performance are: the quality of the detected regions of interest, and the powerfulness of the shape descriptor.

² <http://iapr-tc10.univ-lr.fr/index.php/symbolrecognitionhome>

We would like to briefly report the performance of state-of-the-art approaches that follow the BoVW approach for symbol spotting and retrieval. Luqman et al. [14] used a number of query symbols and 200 line drawings from the same database used here. On this set of images, according to the recall-precision curve obtained from the authors of [14], at recall values between 60.0% and 70.0%, the precision was 60.0%. For recall values above 85.0%, the precision was below 50.0%. Kong et al. [10] used a number of query symbols on 42 line drawings from a different database. According to the recall-precision curve in [10], at recall values between 50% and 70%, the precision was between 62% and 42% respectively.

The approach presented in this paper – although unimproved by spatial verification or post-processing – performs better than those state-of-the-art approaches. As we mentioned in Sect. 2, although our recent work in [17] outperforms the approach presented in this paper, it is rather time and memory consuming. It would be interesting to investigate how to merge ideas from both approaches to achieve both higher accuracy and faster query look-up time.

5 Conclusions and Future Directions

This paper has presented an approach for efficient retrieval of graphical content of line drawings. The approach uses ideas from the famous bag-of-visual-words approach, but presents methods for region detection and description that are suitable to deal with line drawings. The use of simple and standard clustering and learning steps (k-means clustering and SVM learning), suggests that the main two factors that control the retrieval performance are: the quality of the detected regions of interest, and the powerfulness of the shape descriptor.

Future work has many directions. One is performing retrieval from a diverse collection of technical drawings. Another direction is to explore the use of hierarchical representations for clustering the visual vocabulary.

References

1. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2**(3), 27:1–27:27 (2011)
2. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: *Workshop on Statistical Learning in Computer Vision (ECCV)*, pp. 1–22 (2004)
3. Delalandre, M., Pridmore, T.P., Valveny, E., Locteau, H., Trupin, É.: Building synthetic graphical documents for performance evaluation. In: Liu, W., Lladós, J., Ogier, J.-M. (eds.) *GREC 2007. LNCS*, vol. 5046, pp. 288–298. Springer, Heidelberg (2008)
4. Delalandre, M., Valveny, E., Pridmore, T., Karatzas, D.: Generation of synthetic documents for performance evaluation of symbol recognition and spotting systems. *Int. J. Doc. Anal. Recogn. (IJ DAR)* **13**(3), 187–207 (2010)
5. Gersho, A., Gray, R.M.: *Vector Quantization Signal Compression*. Kluwer, Norwell (1991)

6. Honkela, T., Kohonen, T.: Kohonen network. *Scholarpedia* **2**(1), 1568 (2007)
7. Jacobs, D.W.: Robust and efficient detection of salient convex groups. *Pattern Anal. Mach. Intell. (PAMI)* **18**(1), 23–37 (1996)
8. Jacobs, D.W.: Grouping for recognition. Technical report, Cambridge, MA, USA (1989)
9. Jurie, F., Triggs, B.: Creating efficient codebooks for visual recognition. In: *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)*, pp. 604–610 (2005)
10. Kong, X., Valveny, E., Snchez, G., Wenyin, L.: Symbol spotting using deformed blurred shape modeling with component indexing and voting scheme. In: *Graphics Recognition Workshop (GREC)*, online proceedings (2011)
11. Linde, Y., Buzo, A., Gray, R.: An algorithm for vector quantizer design. *IEEE Trans. Commun.* **28**(1), 84–95 (1980)
12. Lloyd, S.P.: Least squares quantization in PCM. *IEEE Trans. Inf. Theor.* **28**, 129–137 (1982)
13. Lowe, D.G.: Object recognition from local scale-invariant features. In: *Proceedings of the International Conference on Computer Vision*, pp. 1150–1157 (1999)
14. Luqman, M.M., Brouard, T., Ramel, J., Llodos, J.: A content spotting system for line drawing graphic document images. In: *International Conference on Pattern Recognition (ICPR)*, pp. 3420–3423 (2010)
15. Nayef, N., Breuel, T.M.: Building a symbol library from technical drawings by identifying repeating patterns. In: Kwon, Y.-B., Ogier, J.-M. (eds.) *GREC 2011*. LNCS, vol. 7423, pp. 69–78. Springer, Heidelberg (2013)
16. Nayef, N., Breuel, T.M.: Statistical grouping for segmenting symbols parts from line drawings, with application to symbol spotting. In: *Proceedings of the 2011 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 364–368 (2011)
17. Nayef, N., Breuel, T.M.: Efficient symbol retrieval by building a symbol index from a collection of line drawings. In: *Document Recognition and Retrieval XX (DRR)*, vol. 8658. SPIE (2013)
18. Nguyen, T., Tabbone, S., Boucher, A.: A symbol spotting approach based on the vector model and a visual vocabulary. In: *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 708–712 (2009)
19. Sivic, J., Zisserman, A.: Video google: efficient visual search of videos. In: Ponce, J., Hebert, M., Schmid, C., Zisserman, A. (eds.) *Toward Category-Level Object Recognition*. LNCS, vol. 4170, pp. 127–144. Springer, Heidelberg (2006)
20. Valveny, E., Delalandre, M., Raveaux, R., Lamiroy, B.: Report on the symbol recognition and spotting contest. In: Kwon, Y.-B., Ogier, J.-M. (eds.) *GREC 2011*. LNCS, vol. 7423, pp. 198–207. Springer, Heidelberg (2013)
21. Yang, S.: Symbol recognition via statistical integration of pixel-level constraint histograms: a new descriptor. *Pattern Anal. Mach. Intell. (PAMI)* **27**(2), 278–281 (2005)