Building a Symbol Library from Technical Drawings by Identifying Repeating Patterns

Nibal Nayef and Thomas M. Breuel

Technical University Kaiserslautern, Germany
nnayef@iupr.com, tmb@informatik.uni-kl.de

Abstract. This paper describes a novel approach for extracting a library of symbols from a large collection of line drawings. This symbol library is a compact and indexable representation of the line drawings. Such a representation is important for efficient symbol retrieval. The proposed approach first identifies the candidate patterns in all images, and then it clusters the similar ones together to create a set of clusters. A representative pattern is chosen from each cluster, and these representative patterns form a library of symbols. We have tested our approach on a database of line drawings, and it achieved high accuracy in capturing and representing the contents of the line drawings.

Keywords: Repeating patterns, Statistical grouping, Shapes clustering, Symbol library, Content analysis.

1 Introduction

A large number of documents contain technical line drawings, such as architectural floor plans, electronic circuit diagrams etc. The users are interested in retrieving specific symbols in a database of drawings. To perform such tasks, the automated content analysis of line drawings is required. Such analysis is essential in applications like retrieval search engines and digital libraries. To this end, we present in this paper our work on analyzing the contents of line drawings and representing them in an indexable representation.

When analyzing images, one finds that they contain objects that consist of components, repetitions of these components in a set of images usually imply that these components are meaningful parts of objects. Hence, identifying those components and clustering the similar ones together, results in a much smaller and a more meaningful representation of a dataset than the images themselves.

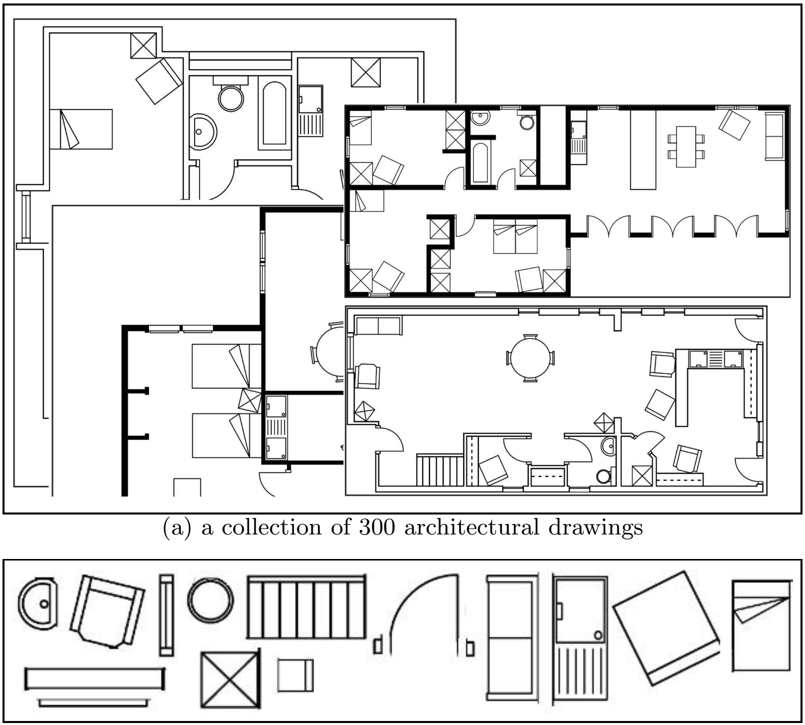
This work discusses a method to extract a library of symbols from a large collection of line drawings. This symbol library is a compact and indexable representation of the complete dataset, which can be used for the development of efficient symbol retrieval systems. The symbol library consists of clusters of symbols patterns from the dataset. The symbols patterns are found using a statistical grouping algorithm introduced by the authors in [8], and the clusters are then formed by clustering the similar symbols patterns. The clustering is based

Y.-B. Kwon and J.-M. Ogier (Eds.): GREC, LNCS 7423, pp. 69–78, 2013.

on geometric matching, which matches the symbols patterns under translation, rotation and scaling.

We show that the method is highly accurate in capturing the contents of the database and in representing it. For capturing the contents of the line drawings, we measure the ability of the algorithm to find all the symbols (or symbols patterns) that appear in the line drawings, and for representing the database as a symbol library, we measure the ability of the proposed clustering algorithm to create the correct clusters of similar symbol patterns.

Figure 1 illustrates the input and output of the proposed method, from a collection of drawings as in Figure 1(a), we get a set of clusters' representatives as shown in Figure 1(b), each representative is a symbol part up to a complete symbol, and each cluster contains all the symbols parts that are similar to each other. We call the set of clusters' representatives a *symbol library.* Using such a library, symbol retrieval becomes straightforward and fast – the retrieval itself is **not** a part of this work though–.


(a) a collection of 300 architectural drawings

(b) a library of symbols and symbols' parts

Fig. 1. Actual results of the proposed method: building a symbol library as in (b) from a collection of architectural drawings as in (a). For clarity of showing, only some drawings are shown, and only a part of the symbol library.

The rest of this paper is organized as follows: Section 2 discusses the related work, Section 3 presents the overall proposed approach. Section 4 presents the experimental results, and in Section 5 we conclude the paper.

2 Related Work

The notion of repeating patterns or visually similar parts in images has been introduced in the literature, but not for complicated line drawings that contain many non-isolated symbols along with connecting background lines.

Most of the works in the literature deal with repetitions within one image. Tuytelaars et. al. [12] presented a Hough transform-based geometric framework for the detection of regular repetitions of planar patterns under perspective skew. Schaffalitzky et. al [11] proposed a RANSAC-based technique for finding the repetitions within a scene. Leung et. al. [6] proposed to find the repeating patterns by matching the neighboring patches in an image under affine transformation using a registration technique. The work of Sanchez et. al [10] discussed using repetitive structured patterns to recognize textured symbols, where they automatically infer an attributed context-sensitive graph grammar to model and recognize a given textured symbol.

Some works have introduced the concept of utilizing repeating patterns across images for retrieval but not in the context of line drawings. A recent work by Doubek et. al. [5] discussed image retrieval using repeating patterns. In their work, the patterns could be found using any of the previously mentioned methods, and then the patterns are described by an invariant descriptor, then matched across images. Another related concept is clustering the visually similar parts in the famous bag-of-(visual)-words approach [2]. This approach is widely used in the field of object recognition and content-based image retrieval. The bag-of-words approach is based on extracting image patches around key points, and coding these patches by transformations-invariant feature descriptors, then clustering the similar descriptors together, this creates a code book for the images parts, and this code book can be later used for object retrieval.

Our approach is conceptually similar to both Doubek et. al. [5] and the works that apply the bag of words approach, however, it uses totally different techniques for each step, since we are dealing with shapes and line drawings. The approach in this paper, applies the concept of repeating patterns to line drawings for the same purpose as the bag-of-words approach. The purpose is creating an indexable representation of a database of images, so that it can be used later for fast retrieval. To the best of our knowledge, there are no published works that utilize repetitions in a database of line drawings.

Our approach has a number of advantages. First, it finds the repetitions in a database of images rather than in individual images as some of the mentioned approaches do. Second, it handles the cases when patterns are not adjacent in an image, or when they are rotated and/or scaled. Third, for finding the patterns, our approach uses a grouping technique [8] that proved to be efficient in extracting meaningful symbol parts from the background. This is advantageous to the methods that use patches or grid regions to extract the patterns. Using patches does not work well for line drawings because the line patterns in a drawing are very similar, which makes the extracted patches not discriminative enough.

3 The Method

In this section, we discuss the two main modules of our approach. The first module deals with extracting a set of patterns from all images, it includes preprocessing and identifying the symbols patterns. The set of patterns is a set of meaningful symbols' parts extracted from the line drawings. The second module takes these patterns and clusters the similar ones together, it uses a geometric matching algorithm as a clustering technique.

3.1 Identifying Symbols Parts via Grouping

The method starts by simple preprocessing of the images of the dataset. First, morphological edge detection is applied, which outputs thin inner and outer contours of the objects in the images. Then a vectorization step is performed by sampling line segments along the contours. We use these line segments as input to the next step that identifies the symbols parts. We call this next step *grouping*, because it groups sets of line segments together to form patterns. The grouping technique has been introduced by the authors in [8].

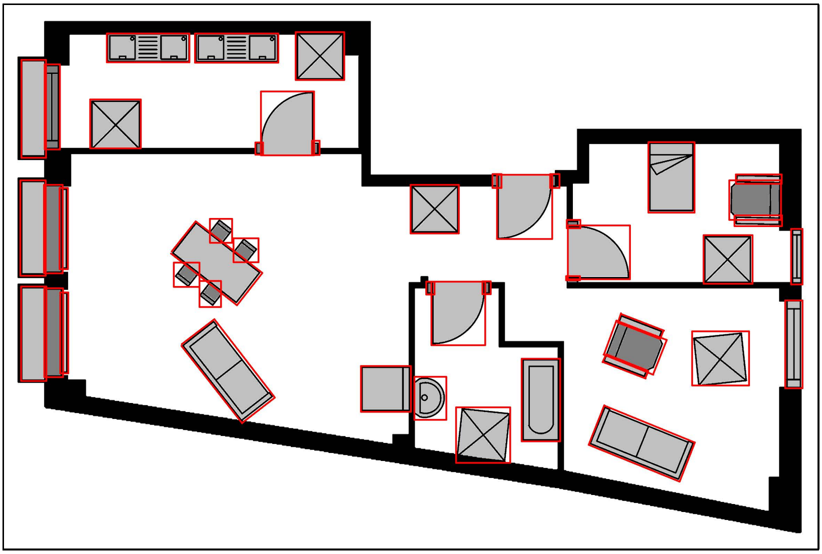


Fig. 2. The found **patterns** of a line drawing (adjacent different patterns have different shading). We draw red bounding boxes around the patterns found by the algorithm.

We apply the grouping procedure on each preprocessed image, and output a set of patterns, where each pattern is a set of line segments. Those patterns correspond mostly to meaningful parts of the symbols of the line drawings. The grouping technique is based on statistical grouping of line segments. It groups a set of line segments together based on non-accidental properties such as convexity. We have shown in [8] that this kind of statistical grouping provides a high probability to capture the symbols parts.

Figure 2 shows the output of the grouping procedure applied on a line drawing. The shaded parts are the patterns – or the symbols parts – found by the procedure. In the figure, red bounding boxes are drawn around the patterns found by the grouping procedure, we also use different shading for adjacent patterns. We define a *pattern* as the set of all the line segments that are inside a group including the segments that constitute the group itself (In Figure 2, this corresponds to all the line segments that lie inside a red bounding box). It is clear that the found patterns of an input image are meaningful parts of symbols up to complete symbols.

For each pattern, we also need to keep information about where this pattern can be found, like which image it comes from and its location in that image.

3.2 Clustering the Patterns via Geometric Matching

Having a list of patterns from all the images of the database, we need to find the repeating – or highly similar – patterns and put them together. Matching the patterns together is done using geometric matching. The patterns that correspond to the same part of a symbol could be rotated or scaled. Recall that each pattern is a set of line segments, so, using the geometric matching algorithm in [1], two patterns can be matched under similarity transformation. Since we are looking for repeating patterns, a pattern is accepted as a matching pattern only if it matches >75% of the other pattern's segments. This is an experimentally-set threshold for clustering the similar patterns together. The following is the clustering algorithm.

```
– clusters: an initially empty list of lists
– patterns: list of all patterns, marked “false” meaning they do not belong to
  any cluster yet.
– for each pattern $p$ of the patterns list
    • if $p$ does not belong to any cluster:
        ∗ add $p$ to clusters
        ∗ use the geometric matching algorithm from [1] to match $p$ with each
          of all other marked “false” patterns
        ∗ add the accepted matches of $p$ to the list of clusters[p] with their
          information
        ∗ mark the accepted matches of $p$ as “true”
    • else: ($p$ already belongs to some cluster):
        ∗ skip p
– End
```

This clustering results in a set of different clusters of patterns. The number of clusters is much smaller than the total number of patterns extracted from all images, hence, this set is a compact representation of the database. Figure 3 shows some of the resulting clusters. We select a cluster representative – i.e. a pattern – from each cluster, the set of the clusters' representatives composes a symbol library.

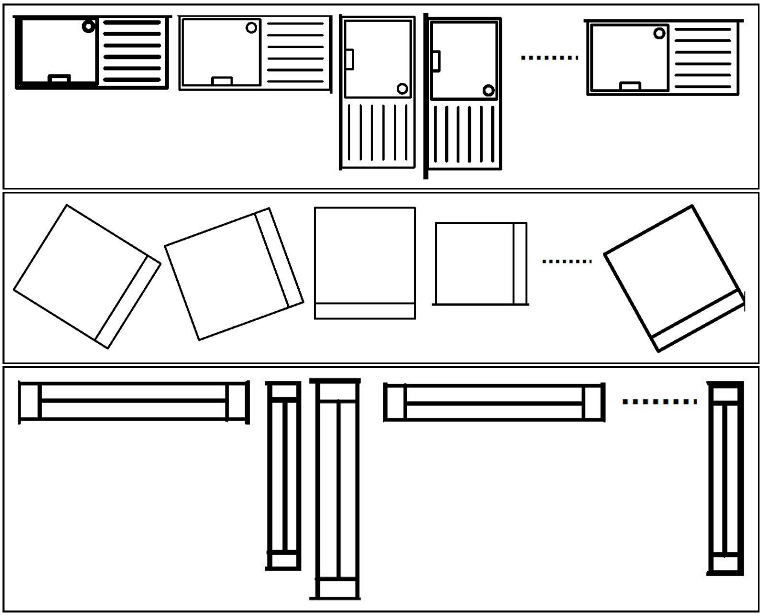


Fig. 3. Clusters: Example clusters of patterns

In this algorithm, there is no need to specify the number of clusters beforehand, the geometric matching step will control the output number of clusters, based on the mentioned acceptance criterion (a pattern should match at least >75% of the other pattern).

It is worth mentioning that the order in which the patterns are examined for clustering, does not affect the results. This is because the geometric matching algorithm [1] we use for matching two patterns will give the same matching score if we match pattern a to patterns b and c or if we match pattern b to patterns a and c, or any other random order.

We should also mention that we do not use soft clustering. That means, in our algorithm, the patterns that are matched with a certain pattern are **not** considered for matching with other patterns. This does not cause problems because we consider the pattern that is currently being matched as the cluster center, or -strictly speaking- the cluster representative, and we match it to the other patterns using a strict acceptance criterion, so, it can be assumed that the matched patterns do not belong to any other cluster. In case of false matching, extra clusters will be formed. It would be useful to investigate how soft clustering or other clustering variants can be used with the proposed algorithm to improve our clustering algorithm.

4 Experiments and Evaluation

4.1 The Dataset

In this section we present the evaluation of the approach on a dataset of architectural drawings. The dataset is a set of 300 images taken from the dataset generated in [3] and in [4], the dataset is a standard dataset in the community of graphics recognition, and it is publicly available[1]. The images are synthesized documents that imitate real complete floor plans with sizes ranging from 2M to 7M pixels. Subsets of this dataset have been used for the current GREC'11 symbol spotting contest, and by researchers for symbol spotting in [9] and [7].

4.2 Evaluation

Here we evaluate the ability of the proposed algorithm to capture the contents of the line drawings, and also its ability to represent those contents compactly as a symbol library. As mentioned previously, using the symbol library for symbol retrieval is not a part of this work, but is a part of a future work.

The performance of the clustering depends on the output of the grouping algorithm, since the patterns found by grouping are the input to the clustering algorithm. We have evaluated the performance of our grouping algorithm on the same dataset in our previous work in [8], where we evaluated the ability of the grouping to find all the parts of the symbols i.e. not missing any parts (recall). Also, whether the found parts are actually relevant symbol parts, not just random segments from different symbols or the background (precision). The grouping algorithm achieved 98.8% recall and 97.3% precision [8].

Now, we present the evaluation of the clustering algorithm. The clustering procedure involves finding the repeating patterns and putting them together. To evaluate the clustering performance, we **adapt** the recall and precision metrics to repeating patterns as follows:

– **cluster recall**: the number of the patterns in the cluster that are similar to the cluster representative divided by the number of all occurrences of that pattern in the dataset.
– **cluster precision**: the number of the patterns in the cluster that are similar to the cluster representative divided by the total number of patterns in the cluster.

Table 1 summarizes the results.

[1] http://mathieu.delalandre.free.fr/projects/sesyd/index.html

Table 1. Results of applying the **CLUSTERING** method to **300** document images of architectural drawings. The provided recall and precision values are the average values for all the formed clusters.

Number of patterns to be clustered	**13780**
Number of the formed clusters	**30**

All Clusters	Avg. Recall	Avg. Precision
	95%	**96.5%**

In Table 1, the "patterns to be clustered", are the patterns found by the grouping module. In some cases, the grouping outputs some irrelevant patterns – non-meaningful symbols' parts –, those irrelevant patterns do not affect the accuracy of the clustering, they only affect the run time, since the clustering procedure has to perform extra matching operations, and also few extra clusters of irrelevant symbols' parts will be formed. Figure 4 shows an example of an extra irrelevant cluster. However, the "recall" of the grouping output which is related to the missing symbols or the missing parts of symbols, affects the "recall" value of the clustering, since the parts missing from the grouping output stay missing from the clusters too.

As Table 1 shows, the clustering module has placed the repeating symbols in clusters with high accuracy. The errors come from unsuccessful matchings. Figure 5 shows an example of a false matching.

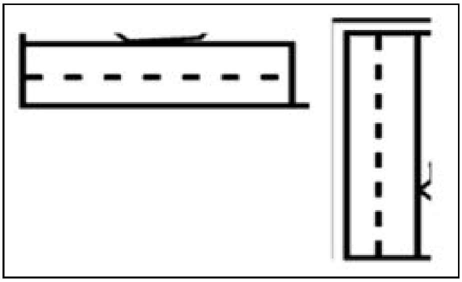


Fig. 4. An example of a cluster that contains irrelevant patterns. Irrelevant patterns are patterns that do not correspond to symbols or parts of symbols.

The running time required to form the clusters is 45 min. on average per forming 1 cluster on a 2.80GHz CPU. This can be significantly improved by speeding up the matching step. However, the running time is still reasonable given the large number of patterns to be clustered (13780) and the achieved high accuracy. The clustering step is to be carried out offline.

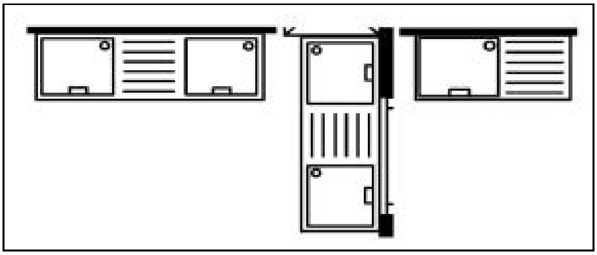


Fig. 5. An example of a cluster that contains a false match. The third symbol (the false match) is very similar to the other symbols that were correctly matched.

4.3 Discussion

In the following, we discuss the importance of clustering the repeating patterns for representing datasets compactly. In a certain application domain like architectural floor plans or electric circuits, there is a defined small set of symbols, which is used to draw the technical line drawings. For example, in the dataset we used for evaluation in this work, only 16 different symbols were used, those symbols appear many times in many of the drawings. The total number of symbols that appear in all the drawings is 6987, and they appear within the drawings connected to other symbols and lines, which means they have to be searched for and located.

Using our proposed approach, we build a library of 30 clusters that contains these symbols and all their repeating instances, along with their location information in the dataset. This library makes the potential applications of fast symbol retrieval from the dataset easy as follows: if we want to retrieve a specific query symbol in this dataset, we only need to match this query to the 30 clusters' representatives, and the best matching cluster will be retrieved. Without the symbol library, we would need to search all the documents in the dataset for a query symbol using a symbol spotting method.

In summary, our proposed approach can be considered as a content analysis method, where a database of document images is processed offline, to get another representation that can later be used for online retrieval.

5 Conclusions and Future Work

This paper has described a novel approach for analyzing the contents of technical line drawings, and has shown some interesting results. Finding the repeating patterns has proved to be an effective way of compactly representing a dataset of technical image documents. The paper also showed the use of feature grouping for extracting meaningful visual patterns. Moreover, the use of geometric matching as a clustering method produced good results for comparing the symbols. Future work includes incorporating the proposed approach in a large-scale symbol retrieval system.

References

1. Breuel, T.M.: Implementation techniques for geometric branch-and-bound matching methods. Computer Vision and Image Understanding (CVIU) 90(3), 258–294 (2003)
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. IJDAR 13(3), 187–207 (2010)
5. Doubek, P., Matas, J., Perdoch, M., Chum, O.: Image matching and retrieval by repetitive patterns. In: ICPR, pp. 3195–3198 (2010)
6. Leung, T., Malik, J.: Detecting, Localizing and Grouping Repeated Scene Elements From an Image. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1064, pp. 546–555. Springer, Heidelberg (1996)
7. Luqman, M.M., Brouard, T., Ramel, J., Llodos, J.: A content spotting system for line drawing graphic document images. In: ICPR, pp. 3420–3423 (2010)
8. Nayef, N., Breuel, T.M.: Statistical grouping for segmenting symbols parts from line drawings, with application to symbol spotting. In: ICDAR, pp. 364–368 (2011)
9. Nguyen, T., Tabbone, S., Boucher, A.: A symbol spotting approach based on the vector model and a visual vocabulary. In: ICDAR, pp. 708–712 (2009)
10. Sánchez, G., Lladós, J.: A graph grammar to recognize textured symbols. In: ICDAR, pp. 465–469 (2001)
11. Schaffalitzky, F., Zisserman, A.: Geometric Grouping of Repeated Elements within Images. In: Forsyth, D., Mundy, J.L., Di Gesú, V., Cipolla, R. (eds.) Shape, Contour, and Grouping 1999. LNCS, vol. 1681, pp. 165–181. Springer, Heidelberg (1999)
12. Tuytelaars, T., Turina, A., Gool, L.V.: Non-combinatorial detection of regular repetitions under perspective skew. PAMI 25(4), 418–432 (2003)