2011 International Conference on Document Analysis and Recognition

Statistical Grouping For Segmenting Symbols Parts From Line Drawings, With Application To Symbol Spotting

Nibal Nayef, Thomas M. Breuel
Technical University Kaiserslautern, Germany
nnayef@iupr.com, tmb@informatik.uni-kl.de

***Abstract*—In this work, we describe the use of statistical grouping for partitioning line drawings into shapes, those shapes represent meaningful parts of the symbols that constitute the line drawings. This grouping method converts a complete line drawing into a set of isolated shapes. This conversion has two effects: (1) making isolated recognition methods applicable for spotting symbols in context, (2) identifying potential regions of interest for symbol spotting methods, hence making them perform faster and more accurately.**

Our grouping is based on finding salient convex groups of geometric primitives, followed by combining certain found convex groups together. Additionally, we show how such grouping can be used for symbol spotting. When applied on a dataset of architectural line drawings the grouping method achieved above 98.8% recall and 97.3% precision for finding symbols parts. Using the grouping information, the spotting method achieved 99.3% recall and 99.9% precision. Compared to the performance of the same method without grouping information, an overall speed-up factor of 3.2 is achieved with the same –or better– recall and precision values.

***Keywords*-convex groups; feature grouping; symbol spotting; document analysis**

I. Introduction

Technical drawings are a major class of document images, and the analysis of those drawings is important for retrieval-by-content search engines and digital libraries. A lot of work has been done on the recognition of isolated symbols, while only few methods have been proposed for spotting symbols in context and/or symbol indexing and retrieval, that is mainly because of the graphical content around a symbol and touching symbols.

There are two approaches to symbol spotting, the first is based on directly describing and indexing regions of interest in the complete drawing, without necessarily recognizing the symbols themselves or segmenting them. The second approach uses some kind of segmentation to get symbols separated from the background, and then recognize those isolated symbols.

Most of the spotting methods in the literature belong to the first approach. This approach has inherent problems, first it is hard to locate zones of interest in a document in a scale-rotation invariant way, second, special descriptors have to be developed for line drawings, as the usual texture-based descriptors won't perform well on the similar local patterns of line segments, third the indexing techniques like hashing do not scale well for large databases. Due to those problems, the methods that implement the first approach result in low precision values. One main reason for following this approach, rather than the segmentation-based approach, is that the current segmentation techniques do not perform well on complex technical drawings.

It is clear, for the methods of the first approach, that if the zones of interest in a technical drawing could be more reliably and precisely located, the later steps of describing and indexing them would achieve better results. As for the methods of the second approach, they always require good prior segmentation.

As a contribution in this direction, we present a grouping method that extracts symbols parts from line drawings, thus converting a line drawing to a group of isolated symbols parts –or regions of interest– that can be later used for spotting symbols, even by isolated recognition methods.
The proposed method applies statistical grouping on vectorial primitives, followed by making combinations of these initial groups to create the final symbol parts. The output parts are seen to correspond to meaningful parts of the symbols up to complete symbols. We also show how this grouping can be used in a symbol spotting method to achieve better results.

The work by Dosch et. al. [1] discusses some grouping mechanisms for line drawings. It is based on studying the relations between pairs of line segments, those relations include collinearity, parallelism and intersections. The relations from local image regions are clustered in buckets. For matching, the signatures of these buckets are compared to the signatures of the symbols models.

As for spotting, various methods were introduced in the literature [2], [3], [4], [5], [6] and [7]. Those methods have introduced some interesting ideas and techniques on various levels. For example, they developed descriptors of zones of interest, whether based on vectorial primitives [2], [6], on graphs [4], [5] or on local shape contexts [3]. Different indexing techniques were used in those works as hashing, relational indexing and inverted file structures.

The rest of this paper is divided as follows. In the next section we explain the grouping technique, in the third section we present the use of this grouping in symbol spotting, after that we show the results of evaluating both

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.81

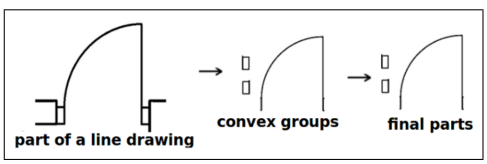
the grouping and the spotting methods, finally we discuss the conclusions and future work.

II. The Grouping Technique

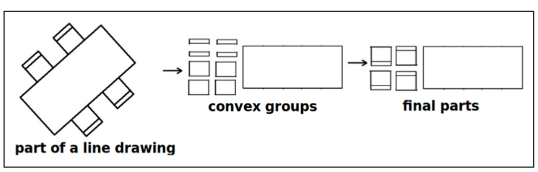
Feature grouping based on non-accidental properties is a computer vision technique that has been used to improve the performance of object recognition [8], [9]. Statistically non-accidental properties of line segments include collinearity, parallelism, co-termination of segments end points, proximity and/or convexity. The segments may be straight or curved, for more details, the reader is referred to [9].

We have chosen convexity as an initial grouping step, because as shown in the statistical analysis of Jacobs [10], that it is unlikely that a random group of line segments will form a convex group. Convexity is an effective grouping property because even though most objects are not convex, they can be decomposed into convex parts. Moreover, convex groups are rotation and scale invariant.

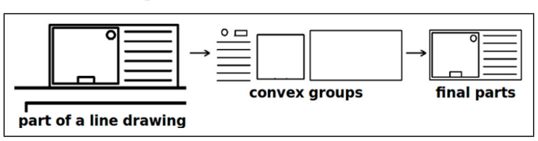
We start by applying simple preprocessing steps on the document images. First, a morphological edge detection step, that produces an image of thin contour lines. Second, a vectorization step is applied simply by sampling line segments along the contours. Those segments are the geometric primitives used as input to the grouping procedure.



(a) The convex groups are the same as the final parts.



(b) Some convex groups are combined to form a smaller number of final parts.



(c) All the convex groups are contained in 1 outer group and form 1 final part.

Figure 1. Grouping: the final parts correspond to meaningful parts of symbols up to complete symbols.

The grouping procedure has the following steps:

1) Apply Jacobs' statistical grouping algorithm [10] for finding salient convex groups, modified to find groups only in counter clock-wise direction
2) Clean up the found groups by removing:
   - Groups that have less than 3 segments
   - Groups with any of the dimensions $>$ x pixels (x=the largest dimension of the largest symbol)
   - Groups that are subsets or cyclic permutations of other groups
3) Keep the convex cycles only (the closed groups), and the open groups that have a series of neither horizontal nor vertical short segments (like arcs)
4) Keep the outer groups, i.e. remove the groups that are completely contained in other groups

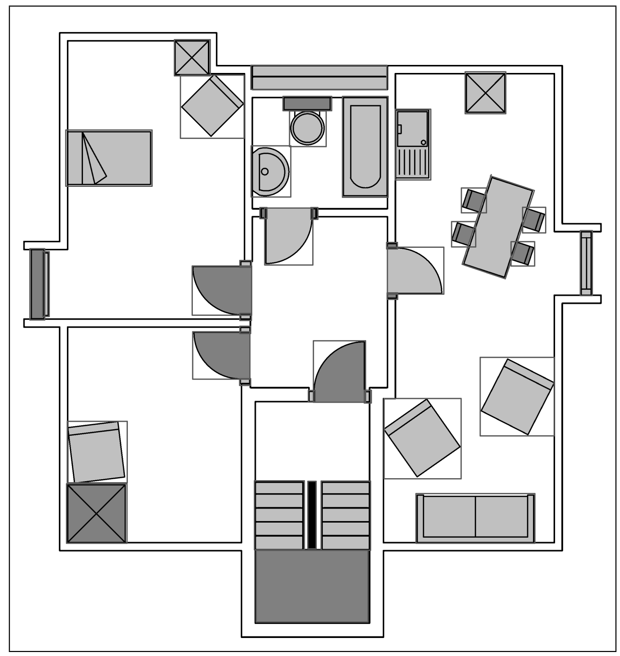


Figure 2. Image parts (adjacent different parts have different shading).

Figure 1 shows how the procedure works on three different symbols.

In this procedure, step 3 is optional and application domain dependent, the dataset is architectural drawings, the architectural symbols consist of closed parts, with very few exceptions like doors symbols. In other technical drawings where more symbols do not consist of closed parts, choosing the salient groups can be based on other non-accidental properties.

Step 4 produces image parts, where an ***image part*** is the group of all segments that are located inside a group including the segments that constitute the group itself. Clearly an image part does not have to be convex. Before step 4, a lot of parts from different symbols are similar, like –for example- the rectangle shape, as noticed in [2], this slows down the later spotting steps and results in a lot of false positives. This step greatly reduces the number of the found groups, and more importantly outputs parts that are different. Figure 2 shows the final output of the grouping module on a complete input image.

As illustrated in Figure 2, the grouping converts a line drawing to a set of meaningful symbols parts, which can be viewed as effective segmentation for the symbols from their interfering background. This process can be considered as an important content analysis step that can be carried out off-line on a collection of line drawings. Another implication of the grouping, is that is converts the hard spotting task into an isolated recognition task.

It is clear now how one might proceed for symbol spotting. One possibility is to compute shape descriptors, or graph representations of symbols parts and then index them, those descriptions would be easier and more accurate to compute on isolated parts than on complete drawings with other graphical context. Another possibility is simply using isolated recognition methods on symbols parts.

III. Symbol Spotting With Grouping

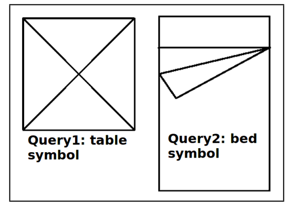
In a previous work we have introduced a technique for symbol spotting and retrieval [11] in real world drawings. This technique was presented for spotting query symbols in context, so, it already does the required job, but here we will show how it can benefit from the grouping information to get faster and more accurate results.

First we review briefly how our previous method [11] works. The method takes a query symbol and a database image as input, then applies the same preprocessing steps discussed in this work –morphological edge detection followed by segments sampling along the edges– to both the query and the image. For finding the matches of the query segments within an image, it performs a branch-and-bound search in the space of the transformations that the query might have undergone to appear in the image. The matching is geometric-based, where a query segment matches an image segment if the transformed query segment aligns with the image segment within a certain error threshold. The method can find multiple instances of the query in an image. For detailed discussions about geometric matching and this method, please refer to [11], [12].

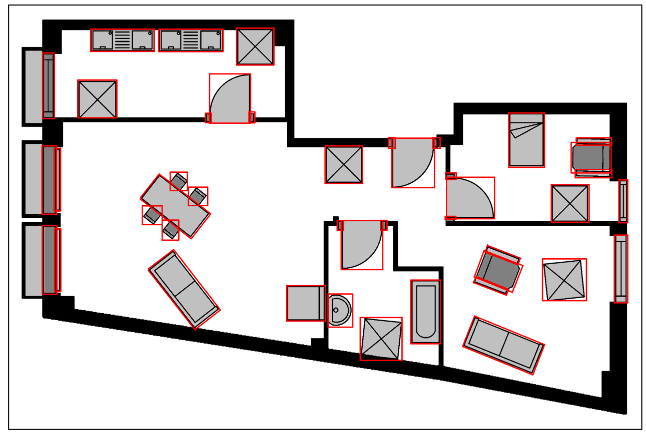
We can incorporate grouping in this method as follows: assume that each image part–that was obtained from the grouping step as shown in Figure 3(b)– is a separate small database image, and then enter all of those small images as input at once, each with its own location information, segments and transformation ranges. In this case, the algorithm would treat those inputs as different initial candidate matches that could have resulted from performing the matching on the complete image.

Even though this way involves applying the matching on many initial images, it would be most of the times faster than matching the query against the complete image, and that is due to three reasons:

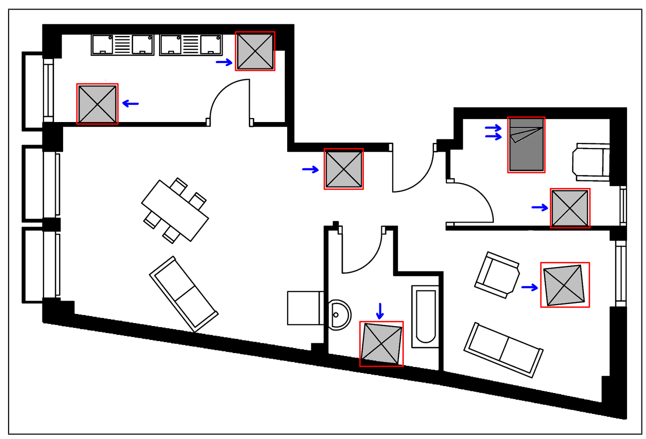
1) The number of segments in the image greatly affects the running time, so for big images with a large number of segments, it is less costly to do the matching on small parts of the image many times –recall that an image part is a group of segments–, than to do it on the whole image once.
2) The running time is also affected by the size of the transformation space, and in small images the vertical and horizontal translation ranges are smaller than them in the complete image.
3) Given that the small images are isolated symbols or parts of symbols, we can compute properties like the scale range and feature weights that can help speed up the matching, and eliminate the false positives.



(a) Query symbols.



(b) The output of the grouping method.



(c) The output of the spotting method.

Figure 3. The complete operation sequence of the spotting method using the grouping method as an intermediate step.

Figure 3 shows the complete spotting operation sequence.

IV. Performance Evaluation

In this section we present the evaluation of both the grouping method and the spotting method on images of architectural drawings. The dataset is a set of 300 images taken from the dataset generated in [13] and in [14], the dataset is publicly available[1]. The images are synthesis documents that imitate real complete floor plans with sizes between 2M to 7M pixels. Subsets of this dataset have been used for GREC'11 symbol spotting contest, and in symbol spotting in [3] and [7].

A. Grouping method evaluation

We will use the recall and precision metrics **adapted to grouping**. Basically, we should check if the grouping module found all the symbols in the drawings, and whether the found parts are actually relevant symbols parts. A symbol is counted as missing if more than 10% of its area is not covered by any of the found parts. And a part of a symbol is counted as irrelevant if it does not represent a symbol part as defined in Section II, for example if the found part consists of random segments from one or more symbols or of segments that do not belong to any symbol.

- **covered symbols recall**: the number of non-missing symbols divided by the total number of symbols in the dataset.
- **covered symbols precision**: the number of relevant found parts divided by the total number of found parts.

Table I
Results of applying the **grouping method** to **300** document images of architectural drawings. **M** is the ground truth total no. of symbols in all images, and **N** is the total no. of parts found by the grouping method.

Number of parts	Ground truth	Results
	12513	**13780**
Covered Symbols	Recall **(M=6987)**	Precision **(N=13780)**
	98.8%	**97.31%**
Avg. running time per image (seconds)		**22.75**

In Table I, the number of found parts (N=13780) is larger than the number of parts that should be found (12513) according to the definition in Section II, but a lot of the extra parts are actually relevant, for example, due to noisy line segments, the same symbol part can be found twice in two slightly different locations, and does not get removed in the cleaning step. Other extra parts are irrelevant, like the rectangles next to the windows symbols in Figure 3(b), they are convex cycles, so the grouping procedure finds them as symbols, and then the subsequent spotting procedure has to perform extra matching operations. So, the "precision" of grouping only affects the running time of the spotting method, but not the accuracy. We disucss the effect of the "recall" of grouping in the next subsection.

Regarding the running time of the grouping method, it took around 23 seconds per image on average. The running time depends on the number of line segments in an image and how many convex groups they can form.

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

B. Spotting method evaluation

For evaluating the spotting method; we used regular recall and precision for retrieval applications. The results are shown in Table II. Note that the spotting method takes its input symbols from the output of the grouping method, hence, if the grouping method missed some symbols, they will not be found by the spotting method. In order to test the spotting performance independently of the "recall" of the grouping method, we have excluded the few queries that correspond to symbols that were not 100% found by the grouping.

In table II, the performance of the method is tested with and without grouping information. In both cases, the same parameters setting is used for accepting or rejecting the matches. As noticed in Table II, the spotting results for the "sofa-1" symbol are lower than the average, that is mainly because it is a simple shape that has few segments, and it is almost a subset of some other symbols. This also causes slower matching with isolated symbol parts.

As for the running time, only the spotting running time is measured, that means we assume the grouping is done off-line as a content analysis step.

The results reported so far in the literature for symbol spotting in general, and for spotting on this particular standard dataset [3], [7], have significantly lower spotting accuracies.

The grouping improves the spotting precision values and helps with the overall speedup. However, as mentioned, the grouping has its own major benefit that is independent of spotting, which is effective segmentation of line drawings and its potential use for content analysis.

V. Conclusion

A statistical grouping method for segmenting technical drawings has been introduced, the method has shown to be effective in describing the content of a line drawing, this description as symbol parts -or complete symbols- is compact and useful for later recognition and spotting steps. Besides, we believe that using grouping of geometric primitives for content description is more suitable for technical drawings than texture based descriptors, as the drawings consist of lines and curves. For future work, we will apply the grouping method on other types of technical drawings and on real world line drawings, and further investigate the use of non-accidental properties for describing image content.

Table II
RESULTS OF APPLYING THE **SPOTTING METHOD** TO **300** DOCUMENT IMAGES OF ARCHITECTURAL DRAWINGS. THE **2nd** COLUMN ENTRIES SHOW THE GROUND TRUTH TOTAL NO. OF INSTANCES OF EACH QUERY SYMBOL IN ALL IMAGES, THEY SUM TO **L=3900**. AND **P1=3860, P2=3972** ARE THE TOTAL NO. OF SPOTTED SYMBOLS WITH AND WITHOUT GROUPING RESPECTIVELY.

Query[a]	Symbol instances	Recall (**L=3900**)		Precision (**P1=3860, P2=3972**)		Avg. time per image (sec.)	
		with grouping	no grouping	with grouping	no grouping	with grouping	no grouping
bed	300	100%	100%	100%	100%	5.3	15.4
table	1172	100%	100%	100%	100%	9.2	19.1
sink-1	201	100%	100%	100%	100%	5.1	15.2
sink-2	112	100%	100%	100%	100%	4.7	88.4
tub	300	100%	100%	100%	100%	27.9	47.8
window[b]	144[b]	100%	100%	100%	100%	10.1	96.0
sofa-1	850	95%	85%	99.6%	80%	23.0	13.0
sofa-2	374	100%	100%	100%	100%	23.0	55.8

[a]We have used only the queries that correspond to symbols that were 100% covered by the grouping algorithm, in order to evaluate the spotting method independently of the grouping's recall values.

[b]For this window query, the results are shown only for 100 images, because in the rest of the dataset, the window symbol is scaled without keeping the aspect ratio, but we only spot similarity-transformed symbols.

REFERENCES

[1] P. Dosch and J. Lladós, "Vectorial signatures for symbol discrimination." in *Graphics Recognition. Recent Advances and Perspectives*, 2003, pp. 154–165.

[2] M. Rusiñol, J. Lladós, and G. Sánchez, "Symbol spotting in vectorized technical drawings through a lookup table of region strings," *Pattern Analysis and Applications*, vol. 13, no. 3, pp. 321–331, 2010.

[3] T. Nguyen, S. Tabbone, and A. Boucher, "A symbol spotting approach based on the vector model and a visual vocabulary," in *Int. Conf. on Document Analysis and Recognition (ICDAR)*, 2009, pp. 708–712.

[4] R. J. Qureshi, J. Ramel, D. Barret, and H. Cardot, "Spotting symbols in line drawing images using graph representations," in *Graphics Recognition. Recent Advances and New Opportunities*, 2008, pp. 91–103.

[5] H. Locteau, S. Adam, E. Trupin, J. Labiche, and P. Heroux, "Symbol spotting using full visibility graph representation," in *Graphic Recognition. Recent Advances and New Opportunities*, 2007, pp. 1–7.

[6] M. Rusiñol, A. Borràs, and J. Lladós, "Relational indexing of vectorial primitives for symbol spotting in line-drawing images," *Pattern Recognition Letters*, vol. 31, no. 3, pp. 188–201, 2010.

[7] M. M. Luqman, T. Brouard, J. Ramel, and J. Llodos, "A content spotting system for line drawing graphic document images," in *Int. Conf. Pattern Recognition (ICPR)*, 2010, pp. 3420–3423.

[8] D. W. Jacobs, "Grouping for recognition," *MIT Artificial Intelligence Laboratory, Memo No. 1117*, 1989.

[9] D. J. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artificial Intelligence*, vol. 31, no. 3, pp. 355–395, 1987.

[10] D. W. Jacobs, "Robust and efficient detection of salient convex groups," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 18, no. 1, pp. 23–37, 1996.

[11] N. Nayef and T. M. Breuel, "Graphical symbol retrieval using a branch and bound algorithm," in *International Conference on Image Processing (ICIP)*, 2010, pp. 2153–2156.

[12] T. M. Breuel, "Implementation techniques for geometric branch-and-bound matching methods," *Computer Vision and Image Understanding (CVIU)*, vol. 90, no. 3, pp. 258–294, 2003.

[13] M. Delalandre, T. Pridmore, E. Valveny, H. Locteau, and E. Trupin, "Building synthetic graphical documents for performance evaluation," in *LNCS: Graphics Recognition. Recent Advances and New Opportunities*, 2008, pp. 288–298.

[14] M. Delalandre, E. Valveny, T. Pridmore, and D. Karatzas, "Generation of synthetic documents for performance evaluation of symbol recognition and spotting systems," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 13, no. 3, pp. 187–207, 2010.