Bag-of-GraphPaths for Symbol Recognition and Spotting in Line Drawings

Anjan Dutta¹, Josep Lladós¹, and Umapada Pal²

¹ Computer Vision Center, Universitat Autònoma de Barcelona, Bellaterra, Barcelona, Spain
{adutta,josep}@cvc.uab.es

² CVPR Unit, Indian Statistical Institute, 203 B. T. Road, Kolkata-700036, India
umapada@isical.ac.in

Abstract. Graphical symbol recognition and spotting recently have become an important research activity. In this work we present a descriptor for symbols, especially for line drawings. The descriptor is based on the graph representation of graphical objects. We construct graphs from the vectorized information of the binarized images, where the critical points detected by the vectorization algorithm are considered as nodes and the lines joining them are considered as edges. Graph paths between two nodes in a graph are the finite sequences of nodes following the order from the starting to the final node. The occurrences of different graph paths in a given graph is an important feature, as they capture the geometrical and structural attributes of a graph. So the graph representing a symbol can efficiently be represented by the occurrences of its different paths. Their occurrences in a symbol can be obtained in terms of a histogram counting the number of some fixed prototype paths, we call the histogram as the Bag-of-GraphPaths (BOGP). These BOGP histograms are used as a descriptor to measure the distance among the symbols in vector space. We use the descriptor for three applications, they are: (1) classification of the graphical symbols, (2) spotting of the architectural symbols on floorplans, (3) classification of the historical handwritten words.

Keywords: Graphic recognition, Symbol recognition, Symbol spotting, Focused Retrieval, Bag-of-GraphPaths.

1 Introduction

Nowadays graphical symbol recognition and spotting has become an important research activity. There is a good demand of effective tools for searching of symbolic objects. The goal is to develop an efficient search engine to find similar graphical objects from a large collection. Shape is an important visual feature and it is one of the basic features used to describe image content. However, shape representation and description is a difficult task, since the real symbol often become corrupted with noise, defects, arbitrary distortion, occlusion etc. Shape representation generally looks for effective and perceptually important features
based on either shape boundary information or boundary with interior content. Various features have been designed, including shape signature, signature histogram, shape invariants, moments, curvature, shape context, shape matrix, spectral features etc. These various shape features are often evaluated how accurately they allow one to retrieve similar shapes from a designated database. However, it is not sufficient to evaluate a representation technique only by the effectiveness of the features employed. This is because the evaluation ignores other important characteristics of a shape representation technique.

There is a long list of shape descriptors available in the literature and majority of them are developed for some specific applications. The major shape descriptors include some simple geometrical, topological properties viz. area, circularity, eccentricity, convexity, ratio of principle axis, circular variance, major axis orientation, bending energy and some combination of them, interested readers are referred to [16] for detailed review. Some of the shape descriptors are based on the points features [1, 3], where the points lie on the boundary of the shape. They use some of the distance measures to get the point to point correspondence between two shapes. Boundary moment of a shape is also used as a shape descriptors. Time series models and especially auto-regressive modeling is also used for calculating shape descriptors [2, 4, 15]. Shape signatures which represent a shape as an one dimensional function derived from the shape boundary points [17] are also used for this purpose. Shapes can be efficiently represented with attributed graphs and for that graph matching, embedding are used to get the similarity measure among graphs representing shapes. The recent major methods work with the graph representation of shapes are [9–11, 14]. Usually graph matching or embedding costs lot computational efforts. Moreover, to cope with real world image distortion, these algorithms include some kind of noise model within it which further increases the cost. So this kind of algorithm works aiming to minimize the computational costs while maintaining the matching efficiency even in the presence of distortion.

In this work we try to explore the power of graphs as a tool for structural representation and for that we propose a graph based shape descriptor for shape recognition. The descriptor is based on the graph representation of the objects where the graph is constructed from the vectorized information of binarized images. Our graph representation considers the critical points detected by the vectorization method as the nodes and the lines joining them as the edges. Our description of a symbol depends on the occurrences of the graph paths. A graph path (Hamiltonian path) between two nodes is the sequence of ordered nodes from the starting node to the ending node. Graph paths capture structural attributes as the topological features. Moreover, graph paths give the serialized representation of graphs, which is efficient in terms of computation. Graph paths also give one type factorized representation which allows tolerance to structural error to a certain level. The occurrences of different graph paths in a symbol contain an important and discriminative structural information which is similar for similar symbols and different for different symbols. The distribution of differ-
ent paths are obtained in terms of a histogram resulted in counting the number of occurrences of each of the paths in a set of prototype paths.

2 Proposed methodology

In this paper we present a descriptor for graphical symbol recognition. The descriptor is based on the graph representation of the objects where the graph is constructed from the vectorized information of the binarized images. Our graph representation considers the critical points detected by the vectorization method as the nodes and the lines joining them as the edges. Our description of a symbol depends on the occurrences of the graph paths. A graph path (Hamiltonian path) between two nodes is the ordered sequence of nodes from the starting node to the ending node. Graph paths capture structural attributes as the topological features. The occurrences of different graph paths in a symbol contain an important and discriminative structural information which is similar for similar symbols and different for different symbols. To capture the path information in each of the symbols, we calculate all the acyclic paths between each of the connected nodes and assign each of them a label of the nearest prototype paths. So at the end we calculate the frequencies of the prototype paths in a symbol. These frequencies create the histogram representation of counting number of prototype paths in a symbol. These histograms capture the distribution of paths in each of the symbols which is discriminative.

2.1 Graph based representation of symbols

To represent an object with a graph we use the binarized images, which again vectorized to get the light weight representation of images. The vectorization process detects the critical points in the binarized images by checking the bending curvature of the point compared to the neighboring points. We consider the critical points detected by the vectorization process as the nodes of graphs and the lines joining them as the edges (see fig. 1).

Fig. 1: The critical points detected by the vectorization method are considered as the nodes and the lines joining them are considered as the edges.
2.2 Bag-of-GraphPaths (BOGP)

An acyclic path between any two connected nodes in a graph is the ordered sequence of nodes from the source node to the destination following the order of the terminal nodes [6,7]. For describing a symbol with the BOGP descriptors, we represent it as a graph and then compute all possible acyclic graph paths between each pair of connected nodes of the graph (see fig. 2). Then all the paths are then described with some shape descriptors. In our case we use the Zernike moments descriptors of order 7 for that purpose, this is experimentally chosen to give the best performance. Let us call the set of descriptors of all the graph paths as \( P_m = \{p_1, \ldots, p_m\} \), and also call a set of prototype paths that are selected by a random prototype selection technique from \( P_m \) as \( P_n = \{p_1, \ldots, p_n\} \) and \( P_n \subseteq P_m, m \geq n \). Then each of the paths in a symbol is assigned as one of the prototype paths using Euclidean distance measure. So at the end it is possible to count the frequency of each of the prototype paths in a graph representing the symbol. This finally represent each of the graphs with a histogram by counting the number of nearest prototype paths occurred in the symbol. Here since the symbol is represented by the count of graph paths and the graph paths are rotation invariant the resulting descriptors of the symbols are also rotation invariant.

![Fig. 2: Different acyclic paths between each pair of connected nodes.](image)

A descriptor for a symbol \( S_1 \) with the set of paths \( P_{S_1} = \{p_1, \ldots, p_n\} \) is the histogram of counting the number of paths similar to each of the prototype paths:

\[
BOGP(S_1) = [\#p_1 \in P_{S_1}, \ldots, \#p_n \in P_{S_1}] \quad (1)
\]

These descriptors contain similar distribution for similar symbols and different distribution for different symbols (see fig. 3).

3 Experimental results

3.1 Symbol matching

In order to evaluate the proposed methodology, we present a symbol matching experiment. The set of prototype paths is created by randomly selecting 1000 paths from the input data, which results in the dimension of the BOGP vector as 1000. The BOGP descriptors are computed for all the symbols in which we perform symbol matching and when we get the query symbol we also compute the descriptor and measure the Euclidean distance to get the ranked list of symbol
for a particular query. The smaller the distance is, more similar the symbols are. We use two different isolated symbol dataset for that purpose and they are: (i) SESYD Queries (floorplans) [5] and (ii) GREC-POLY [12] (see fig. 4). Both of these two datasets are created with some kind of noise and distortion model to simulate the noise introduced by the real world situation. We compute the precision ($P$), recall ($R$) and average precision ($\text{AveP}$) of the retrieval list to get the quantitative measure of the results. The readers are referred to [13] for further details about those measures in information retrieval.

![Image](image.png)

Fig. 3: Histogram to compute the number of paths is shown for similarity measure. BOGP descriptors give similar distribution for similar symbols and different distribution for different symbols.

**SESYD Queries (floorplans)** This dataset contains synthetically generated corpus of symbols cropped from complete documents. These experiments are focused on evaluating the robustness of the proposed algorithm against the context noise i.e. the structural noise introduced in symbols when they are cropped from the documents. We believe that being successful on this kind of noise is very important when the algorithm is intended to apply for symbol spotting on a whole document. This dataset contains 3 levels of difficulties of structural noise each level containing 1000 images results in 3000 floorplan symbols in total and 16 ideal symbols used as the model symbols.

We got 60.79% precision ($P$), 80.58% recall ($R$) and 85.26% average precision ($\text{AveP}$) on this dataset, which indicates the success of the proposed algorithm. But in case of symbols having similar substructures the algorithm confuses, this explains the small amount of errors in the results. But we got good average precision which shows the true positives occur at the beginning of the ranked list.
GREC-POLY This dataset is mainly adapted from the symbol recognition contest of GREC,'2005. The bitmap images are degraded with the noise model proposed by Kanungo et al [8], which simulates the datasets with the noise introduced by the scanning process. The authors have applied three separate sets of parameters to generate three different degradation levels, where each of the total 150 model symbols are degraded to generate 300 degraded images, which results in total 45000 isolated symbols. The dataset is available in vectorized form which is proceeded by a simple morphological operation and a connected component analysis to label the closed regions and the internal and external contours composing a symbol. This dataset also contains arbitrary rotation and scaling.

For this dataset, we got 78.76% precision ($P$), 88.67% recall ($R$) and 93.83% average precision ($AveP$). The results clearly show the efficiency of the method. The method is more successful in this dataset, because it creates the descriptors based on the factorized substructures of graph representation of the symbol which can efficiently tolerate the deformation that this dataset contains.

3.2 Symbol spotting

This experiment is done to show the effectiveness of the proposed descriptors for spotting symbols on documents. We use the SESYD floorplan database [5] for the experiment purpose. This is a collection of synthetically generated images. The database contains 10 sub-database, each of the sub-database contains 100 floorplans. Each floorplans of a sub-database is created by randomly placing different architectural symbols on a fixed floorplan template in different scales and orientations.

For spotting or detection we need some localization technique, for that we run sliding window of size $30 \times 30$ pixels with 60% overlapping and capture the
path information in the window. Then a BOGP histogram is computed for each window. The set of prototype paths is created by taking the 1000 randomly selected paths from the given documents. So in this case each of the sliding windows result in a BOGP histogram vector of size 1000. Since spotting intended to work with large dataset, searching of the query should be more efficient. For that we organize the descriptors of each of the window in hash tables with locality sensitive hashing (LSH) technique. So when the query is invoked, it is described by a BOGP descriptors and then looking into the hash tables results in set of retrievals. The retrievals are then ranked in ascending order based on the distance measure. The retrievals at beginning of the list are supposed to be more relevant to the queried one.

3.3 Handwritten word recognition

It is obvious that any graphical object can be given symbolic representation. This experiment is done to check whether our BOGP descriptor is also eligible to capture the information of handwritten words. For that we choose to apply the descriptor to a word recognition scenario. We use a corpus contains 27 pages from a collection of marriage registers from the Barcelona Cathedral, where all the pages are well segmented up to the word level. To apply our descriptor we consider each of the segmented words as a symbol and use the same experimental settings as of sub-section 3.1. In this case also the set of prototype paths is created from 1000 randomly selected paths from the given data. The method needs a preprocessing step including binarization which is done by the Otsu algorithm.

Query:

![Query Image]

Results:

![Results Image]

Fig. 5: Qualitative results for the Barcelona Cathedral collection by our method for the word "Farrer".

The precision (P), recall (R) and average precision (AveP) attend by the system are respectively 0.65%, 93.77% and 9.73%. Although the quantitative results seem to be quite bad, the results contain visual similarity with the queried word. In fig.5 we present a qualitative result for a given query word "Farrer" for
the Barcelona Cathedral collection. We can see that the method presents some false positives in the first ten responses. In general, handwritten words are very cursive in nature. So in a small region where the curvature of the writing is high, it creates a lot of spurious points. So small variation of in the writing styles creates lot of difference after they are represented by the graphs. This can explain the bad results in case of the handwriting recognition.

4 Conclusions and future works

In this paper we have presented a symbol description technique based on the graph representation of objects. The descriptor is based on the distribution of the graph paths of a graph representing the symbols. Graph paths contain structural attributes as features and the distribution of those paths in a similar symbol is similar but different for different symbols. We tested the methods on the recognition of isolated symbols and for symbol spotting, the results are encouraging.

We also investigated whether we could apply our descriptor for representing handwritten words but in that case our results are not as expected. This is due to the cursive nature of the handwritten words, whose little variance creates lot of difference in the our graph representation. So the future work will concentrate on introducing a more efficient noise model to the existing system.

Acknowledgement

This work has been partially supported by the Spanish projects TIN2009-14633-C03-03, TIN2008-04998, CSD2007-00018 and the PhD scholarship 2011FI_B01022 by the AGAUR.

References


