New Approach for Symbol Recognition Combining Shape Context of Interest Points with Sparse Representation

Thanh Ha Do, Salvatore Tabbone, Oriol Ramos Terrades

To cite this version:

Thanh Ha Do, Salvatore Tabbone, Oriol Ramos Terrades. New Approach for Symbol Recognition Combining Shape Context of Interest Points with Sparse Representation. 12th International Conference on Document Analysis and Recognition ICDAR 2013, Aug 2013, Washington, DC, United States. hal-00939182

HAL Id: hal-00939182
https://inria.hal.science/hal-00939182
Submitted on 30 Jan 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
In this paper, we propose a new approach for symbol description. Our method is built based on the combination of shape context of interest points descriptor and sparse representation. More specifically, we first learn a dictionary describing shape context of interest point descriptors. Then, based on information retrieval techniques, we build a vector model for each symbol based on its sparse representation in a visual vocabulary whose visual words are columns in the learned dictionary. The retrieval task is performed by ranking symbols based on similarity between vector models. The evaluation of our method, using benchmark datasets, demonstrates the validity of our approach and shows that it outperforms related state-of-the-art methods.

I. INTRODUCTION

Symbol recognition system are composed of two phases: symbol description and symbol retrieval. The description phase consist of defining local shape descriptors, invariant to similarity transforms, robust to local symbol distortions and robust to document noise. The retrieval phase consist of implementing matching algorithms together with indexing, or hashing, techniques to effective retrieve queried symbols. This paper focus in the symbol description phase, while symbol retrieval relies on state-of-the-art information retrieval techniques that have proven to perform well in video retrieval as well symbol retrieval [17].

Regarding to shape descriptors and based on the objective of encoding the shape of a symbol, Marçal et al. [13] divide description techniques into three different main categories. These categories are photometric description, geometric description and syntactic and structural description. Photometric description is suitable for recognizing complex symbols, e.g., logos. Techniques in this category include SIFT descriptor [10], moment-based descriptors [4], [5], [8], and generic Fourier descriptor (GFD) [26]. SIFT descriptor characterizes each interest point by the local edge distribution around the point. This is very useful to describe complex symbols, but it loses effectiveness when representing simpler ones (see Section IV). Moment-based descriptors have some advantages in symbol description comparing to other descriptors. For instance, they are invariant to translation, scaling, and rotation transforms. In addition, we can recover original symbols from moment descriptors [5], [8]. However, they still have some shortcomings, e.g., do not allow multi-resolution analysis of a shape in radial direction [27]. GFD is extracted from spectral domain by applying 2-D Fourier transformation on polar shape symbol. Hence, it overcomes the problem of noise sensitivity while still be invariant under affine transformations.

Geometric description techniques are primitive-based methods that have shown to be useful when the symbols are non-isolated and be affected by occlusions [15], [16], [19]. Some examples of such primitives are contours, closed regions, connected components, skeletons, etc. to enumerate some of the most popular ones. Although these descriptors can easily be computed, they are usually poorly discriminant [16], and sometimes the matching process is time consuming [15]. In addition, there are also several primitives describing symbols using geometric information [12] or vector signatures [6]. In general, they are invariant under similarity transformations, but depend on a prior normalization step to achieve invariance and are very sensitive to noise at vector level.

There has been a great number of works on finding good syntactic and structural descriptions [3], [9], [14], [24]. These descriptors aim at defining relationships between primitives. In [3], [14], the authors propose rule-based descriptions whose performance are highly affected by noisy data. In [9], [24], the authors propose structural descriptors which present symbols as one-dimensional string or by an attribute relational graph. In general, graph-based descriptors are powerful tools for symbol description but the computation time linked to them is huge since we have to deal with sub-graph isomorphism, which is NP-hard. In summary, most of photometric descriptors requires well-segmented symbols to satisfactory perform while geometric descriptors have low discriminant capacities and structural descriptors are computationally demanding.

In this paper, we extend the work done in [17] by providing a sparse representation of local descriptors based on keypoints. Sparse representation has been widely used in image denoising, separating, and extracting text regions [7], [28].
But, to the best of our knowledge, sparse representation has not been applied to symbol descriptors. In this way, we achieve a sparse description of invariant descriptors which will improve the retrieval performances.

More specifically, we first compute shape context of interest points in symbols and use them as training dataset for learning a sparse dictionary by means of the K-SVD algorithm [7]. Then, we consider the learned dictionary as a visual vocabulary whose visual words are each of the entries of this dictionary. Next, we construct a vector model for every symbol based on its sparse representation in the vocabulary and we adapt the tf-idf approach to the sparse representation. Finally, the retrieval task is performed by ranking symbols based on their similarity with the query where the similarity is computed based on the vector model approach.

We have organized the rest of this paper as follows. In Section II, we present some fundamental background on shape context and shape context of interest points descriptors. Our proposed method and retrieval model are presented in Section III and we report experimental results in Section IV. Finally, we conclude and discuss the future work in Section V.

II. RELATED WORK

In this section, we present the main background for shape context and shape context of interest point as well as their main invariant properties under rotation and scaling transforms.

A. Shape Context

Shape context (SC) is one of the descriptors with higher accuracy rates in many shapes recognition tasks and was introduced in [2]. Shape boundaries, either internal and external, are sampled in n points. For each point \( p_i \) on the symbol contour, Belongie et al compute its coarse histogram \( h_i \) of the relative coordinates of the remaining \( n-1 \) points:

\[
h_i(l) = \text{card}\{c \neq p_i : (c-p_i) \in \text{bin}(l)\}, l = \overline{1,L}
\]

where \( c \) are contours points expressed in log-polar coordinates and \( L \) is the number of bins of the SC histogram at point \( p_i \). Thus, for each symbol S, its SC is the real matrix \( H = \{h_1, \ldots, h_n\} \) with dimensions \( L \times n \).

Since all measurements are computed with respect to all points that are sampled from internal or external contours on the symbol, SC is therefore invariant under shape translation. Invariance under scaling is obtained by normalizing all radial distances by the mean distance among all point pairs in the symbol. Moreover, it is inherently insensitive to small perturbations of symbols, and indeed it is robust to small nonlinear transforms.

B. Shape Context of Interest Points

SC defined so far show two main drawbacks when applied to symbol retrieval tasks. On the one hand, as most of photometric descriptors, we need to segment symbols well enough for having satisfactory retrieval performance. On the other hand, matching function is computationally time-demanding if the number of boundary points is large.

Inspired by the works of detecting efficiently an object from its key-points (also known as interest points) [1], [11], the authors in [17] proposed a new approach, named Shape context of interest point (SCIP). In their approach, SC is only defined around interest points. More specifically, interest points \( IP = \{p_1, p_2, \ldots, p_r\} \) and contour points \( C = \{c_1, \ldots, c_n\} \) of a given symbol are detected. Indeed, each of these interest points \( p_i \) is thus a reference point to compute the SC of a symbol. Because the \( IP \) set is rarely a subset of \( C \) for most of the cases [20], the same rotation normalization method for SC cannot be applied to SCIP. Instead, dominant orientation of interest point information for orientation normalization is used. In more detail, each interest point \( p_i \) is represented by its coordinates and the dominant orientation: \( p_i = \{x_i, y_i, e_i\} \).

The relative log-polar coordinates of contour points \( c_j \in C \) is denoted by \( c_j = (\log r_{ij}, \theta_{ij}) \) in which \( r_{ij} \) is the normalized distance from \( p_i \) to \( c_j \), and \( \theta_{ij} = \langle p_i c_j, e_i \rangle \). The coarse histogram at \( p_i \) is computed as:

\[
h_i(l) = \text{card}\{c_j \neq p_i : (c_j-p_i) \in \text{bin}(l)\}, l = \overline{1,L}
\]

Then, the SCIP descriptor is the set \( H = \{\tilde{h}_1, \tilde{h}_2, \ldots, \tilde{h}_r\} \), where each \( \tilde{h}_i \) is a histogram of \( L \) bins.

III. SPARSE REPRESENTATION AND SYMBOL RETRIEVAL

Querying symbols on a dataset using SCIP descriptor needs to assign accurately each SCIP descriptor to a visual word. In the proposed approach we avoid this step by describing each SCIP descriptor by a linear combination of visual words being each entry of a learned dictionary \( A \). In this section, we will first explain how to learn a dictionary from a set of SCIP descriptors providing sparse representation and then, how to built vector models permitting to us querying symbols in a dataset.

A. Learned Dictionary of SCIPs

An over-complete dictionary \( A \) for sparse representation is a dictionary built from a family of training signals in which each signal has an optimally sparse approximation in \( A \).

In this paper, we use SCIPs \( \{\tilde{H}_n\}_{n=1}^N \) extracted from a set of \( N \) training symbols as a family of training signals. Each training signal \( h_i \in \mathbb{R}^L \) has an optimally sparse approximation \( x_i \) satisfying \( \|h_i - Ax_i\|_2 \leq \epsilon \) with \( h_i^* = Ax_i \).

\[
\min_{x_i,A} \sum_i \|x_i\|_1 \text{ subject to } \|h_i - Ax_i\|_2^2 \leq \epsilon
\]

Such a dictionary can be obtained by solving the problem defined in Equation 3. To do this, Aharon et al [7] proposed a 2 steps iterative algorithm called K-SVD. In this algorithm, they iteratively adjust \( A \) via two main stages: sparse coding stage and update dictionary stage. In the sparse coding stage, all sparse representations \( X = \{x_i\} \), of \( Y = \{\tilde{h}_i\} \), are computed while keeping \( A \) fixed. These sparse representations can be computed by an algorithm that approximates the solution of Equation 4. The algorithm used by the authors, and also by us
in this approach, is the orthogonal matching pursuit (OMP)
algorithm [23].

\[ x_i = \min_x \| x \|_0 \text{ subject to } \| \hat{h}_i - Ax \|_2^2 \leq \epsilon \]  

(4)

In the update dictionary stage, an updating rule is used to
optimize the sparse representations of the training signals.
In general, the way to update the dictionary is different from
one learning algorithm to another. In K-SVD algorithm,
the updating rule is applied column-wise on the dictionary’s
matrix \( A \). Thus, each column \( a_{i0} \) of \( A \) is updated sequentially
such as minimizing the residual error in Equation 5:

\[ \| Y - AX \|_F^2 = \| (Y - \sum_{i \neq i0} a_ix_i^T) - a_{i0}x_{i0}^T \|_F^2 
\]

\[ = \| E_{i0} - a_{i0}x_{i0}^T \|_F^2 \]  

(5)

since all columns in \( A \) other than \( i_0 \)-th column are fixed, \( E_{i0} \)
is also fixed. Thus, the minimization of Equation 5 depends
only on the optimal \( a_{i0} \) and \( x_{i0}^T \), where \( x_{i0}^T \) refers to the \( i_0 \)-th row of \( X \). This problem is therefore converted to a problem
of approximating a matrix \( E_{i0} \) by a rank 1 matrix by minimizing
the Frobenius norm. Moreover, to ensure the sparsity in vector \( x \), \cite{7}
defined a group of indexes \( \Omega_{i0} \) where \( x_{i0}^T \) is different to zeros,
and \( E_{i0} \) is the \( E_{i0} \) matrix restricted to those indexes.
Then, the minimization of Equation 5 is equivalent to the
minimization with respect to \( x_{i0} \) of the equation:

\[ \| E_{i0}\Omega_{i0} - a_{i0}x_{i0}^T\Omega_{i0} \|_F^2 = \| E_{i0} - a_{i0}x_{i0}^T \|_F^2 \]  

(6)

The optimal solution \( x_{i0}^R \) in Equation 6 has the same support
as the original \( x_{i0}^T \), and can be found by calculating the singular
value decomposition (SVD) of the error matrix \( E_{i0} \).

B. Visual vector model

After applying the K-SVD algorithm on the set of SCIPs
descriptors \( \{ \hat{H}_k \}_{k=1}^N \) used as training dataset, we have learned
a dictionary \( A \in \mathbb{R}^{L \times K} \) as well as the sparse representations
of all SCIPs in that dataset. In the remainder of this section
we will explain how we can build up a visual vector model
from the sparse representation of SCIP descriptor that we will
use for retrieval tasks in the experimental sections.

Without loss of generality, we can assume that \( \hat{h} \in \mathbb{R}^L \)
is one SCIP descriptor in \( \{ \hat{H}_k \}_{k=1}^N \) and \( x \in \mathbb{R}^K \) is the sparse
representation of \( \hat{h} \) given the dictionary \( A \). Instead of assigning
a single visual word, typically the nearest centroid of a cluster
given by a k-means algorithm like in \cite{17, 18}, \( \hat{h} \) can be seen
as a linear combination of the columns of the dictionary \( A \).
Therefore, we have to adapt the vector model construction to
the sparse representation of SCIP descriptors.

Let \( I \) be the indexes set where \( x \) is different to 0. We define the characteristic vector \( v \in \mathbb{R}^K \) as being the
0-1 valued vector \( v(k) = 1 \) if \( i \in I \) and 0 otherwise.
For example, if \( I = \{ 1, p - 1, p, q, q + 1 \} \) then \( v = \{ 1, 0, \ldots, 1, 0, \ldots \} \).
For each training symbol \( n \), \( \hat{H}_n = \{ \hat{h}_1^n, \hat{h}_2^n, \ldots, \hat{h}_{r_n}^n \} \) is
its SCIP descriptor set and \( r_n \) is number of interest points
detected for such symbol. Then, \( v_{i0}^{n} \) denotes the characteristic
vector of \( \hat{h}_i^n \), which is defined by the sparse representation \( x_{i0}^n \)
of \( \hat{h}_i^n \) given the learned dictionary \( A \). Thus, the columns of the
dictionary matrix \( A \) will play the role of words in a visual
word vocabulary framework.

Similarly to the tf-idf approach used in information retrieval
for building vector models, we define \( t_f \) and \( idf \) factors to
describe, respectively, the document contents and the impor-
tance degree of terms. Herein, documents and symbols are
identified. Thus, \( f_k^n \) is the frequency of the word \( k \) in a
symbol \( n \) and \( t_{fk,n} = \frac{f_k^n}{\max_k f_k^n} \). Observe that we can easily
computed these frequencies through the characteristic vector:

\[ f_k^n = \sum_{i=1}^{r_n} v_{i0}^{n}(k). \]

The idf factor is similarly defined as usual on information
retrieval systems but also adapting its definition to the sparse
representation of SCIP descriptors. The importance in distin-
guishing a relevant symbol from non-relevant one in a database
is measured by \( \log \frac{N}{f_k^n} \), where \( l_k \) is the number of symbols in
which the word \( k \) appears.

\[ l_k = \text{card} \{ n = 1, N | f_k^n \neq 0 \} \]

Therefore, the vector model for a given symbol is defined
by the weighted frequency for all words \( k \) in our visual
vocabulary:

\[ w_{q}(k) = t_{f_{k,n}} \times idf_{k} = \frac{f_{k,n}}{\max_k f_k^n} \times \log \frac{N}{l_k} \]  

(7)

C. Retrieval Symbol

For each query symbol \( s_q \) in the set of query symbols
\( S_q = \{ s_1, s_2, \ldots, s_m \} \), its vector model is computed in
the same way described in section III-B. We first compute its
SCIP descriptor \( \hat{H}_q = \{ \hat{h}_1, \ldots, \hat{h}_{m_q} \} \). Then, using the
learned dictionary \( A \), we find the sparse representation of each
element in \( \hat{H}_q \) by applying the OMP algorithm \cite{23} to solve
Equation 4. Finally, we compute the vector model, named \( w_q \),
as summarized in Equation 7.

Next, the similarity of a query symbol \( s_q \) in a database \( s_n \subseteq S \) is computed as the cosine distance between two vectors \( w_q \) and \( w_n \):

\[ \text{distance}(w_q, w_n) = \frac{\langle w_q, w_n \rangle}{|w_q| \times |w_n|} \]  

(8)

where \( \langle \cdot, \cdot \rangle \) is the dot product. Finally, symbols in the database
are ranked based on their similarity to the query symbol \( s_q \).

IV. EXPERIMENTAL RESULTS

We have evaluated our method with the GREC2 synthetic
benchmark. This dataset contains occurrences of 50 different
symbols obtained by linear transforms (rotation and scaling)
and by applying deformation and degradations processes. We
have used 50 symbols as queries to retrieve similar symbols
to them in the following subsets: dataset \( D_1 \) includes 250
symbols generated by linear transforms (rotation and scaling);
dataset \( D_2 \) includes 75 symbols generated by strong non-rigid
transforms; and dataset \( D_3 \) includes 75 symbols generated by

\[^2\text{http://www.cvc.uab.es/grec2003/SymRecContest/}\]
strong non-rigid transforms and Kanungo noise. The number of occurrences for each class is not the same for all of them: ranging from 1 to 10 times in dataset 1; from 1 to 11 in dataset 2 and ranging from 0 to 11 in dataset 3. We use the set \( D_i, i = 1, \ldots, 3 \) to construct the learned dictionary using K-SVD algorithm where the number of columns \( K \) is 512.

To examine the effectiveness of the proposed descriptor, we have compared it to 6 state-of-the-art methods for symbol recognition, namely R-signature [21], GFD [25], Zernike moments [22], SIFT [11], SC [2] and SCIP [17]. In addition, for Zernike moments descriptors, we have built two descriptors. The first descriptor, \( G_1 \), includes 32 low-order moments while the second one, \( G_2 \), includes 32 high-order moment. We have only considered the magnitude of Zernike moments for both descriptors \( G_1 \) and \( G_2 \) and for each one, the computed moments satisfy the following conditions:

\[
G_1 = \begin{cases} 
3 \leq n \leq 10 \\
|m| \leq n \\
n - |m| = 2k \\
k \in N 
\end{cases} \quad G_2 = \begin{cases} 
10 \leq n \leq 17 \\
|m| \leq n \\
n - |m| = 4k \\
k \in N 
\end{cases}
\]

For SIFT, SC and SCIP descriptors we have used k-means algorithm to build the visual dictionary, where each cluster centroid is a visual word. The number of clusters is 175 for the three descriptors. The similarity measure for retrieval tasks is always the cosine distance, as defined in Equation 8. We have used precision-recall rate, denoted by \( R \) and \( P \) respectively, to evaluate the retrieval task:

\[
R = \frac{\text{the relevant symbols retrieved}}{\text{the relevant symbols existing in the database}} \\
P = \frac{\text{the relevant symbols retrieved}}{\text{the number of retrieved symbols}}
\]

Finally, we have conducted two experiments to assess the goodness of the proposed approach. The first experiment aims at finding the best size of SCIP descriptor (\( L \)) while the second experiment aims at evaluating the retrieval performance.

For the first experiment, devoted to find the best dictionary corresponding with the size of SCIP (\( L \) value), we have compared the averages precision and recall rates performed on dataset \( D_1 \) for the number of bins \( \log(r) \) ranging from 3 to 5 and the number of bins for \( \theta \) is set to 12 and 16, respectively. Therefore, the descriptor dimension \( L = \theta \times \log(r) \) belongs to \( \{36, 48, 60, 64, 80\} \). Figure 1 shows that we have obtained the best precision and recall rates when \( L = 12 \times 3 \) or \( L = 16 \times 3 \).

We show results for the second experiment in Figure 2. As it was also observed in [17], we can observe that SIFT descriptor loses its effectiveness when applied to symbols. SCIP descriptor have proved to be more suitable for symbol retrieval than SIFT in technical documents. Building a visual vocabulary based on sparse representation provides better results than using cluster algorithms like k-means and assigning just one visual word to each local descriptor. Results also show that our approach outperforms the compared methods in datasets \( D_1 \) and \( D_2 \) while for the \( D_3 \) dataset GFD show better performance. This result can be explained by the fact that the key-point detection step is sensitive to noises.

Some retrieval examples are giving using our method in Figure 3. Thanks to the SCIP descriptor and sparse representation, we can remark that our approach is robust to scale and rotation.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a new approach for symbol description based on SCIP descriptors and sparse representation. This new approach is invariant under rotation, scaling, and distortion, since SCIP descriptor is. Also, it is well-adapted to degraded and noisy symbols since sparse approaches are robust to this kind of degradations.

Obtained results for benchmark dataset have proven that our descriptor is suitable for symbol description for retrieval task and competitive to related state-of-the-art methods. Indeed, by describing each SCIP descriptor as a linear combination of visual words we have achieved better system performance.

In the future, we would like to apply this approach to symbol spotting problem in large graphical documents where symbols can not be easily segmented.

REFERENCES


Fig. 1. The effect of \( L \) on the precision and recall rate.

![Figure 1](image-url)
Fig. 2. Retrieval effectiveness with recall/precision rates in datasets: $D_1$ (left), $D_2$ (center) and $D_3$ (right). The legend is the same for all plots.

Fig. 3. Some retrieval examples in $D_1$: the query symbol is in the first column; other columns are the nearest matches ranked from left to right.