Sparse Representation over Learned Dictionary for Symbol Recognition

DO Thanh Ha, Salvatore Tabbone, and Oriol Ramos Terrades

Abstract

In this paper we propose an original sparse vector model for symbol retrieval task. More specifically, we apply the K-SVD algorithm for learning a visual dictionary based on symbol descriptors locally computed around interest points. Results on benchmark datasets show that the obtained sparse representation is competitive related to state-of-the-art methods. Moreover, our sparse representation is invariant to rotation and scale transforms and also robust to degraded images and distorted symbols. Thereby, the learned visual dictionary is able to represent instances of unseen classes of symbols.

Key words: Symbol Recognition, Sparse Representation, Learned Dictionary, Shape Context, Interest Points.

1 Introduction

The increasing number of digital images along with the computation power of mobile devices have boosted to a renewed interest in many computer vision tasks such as content-based image retrieval (CBIR), image understanding and object recognition. In the field of document analysis, symbol retrieval is even a more challenging task since images usually are gray-scale, if not black and white, and shape information is the only source of information available. Moreover, symbols represent an abstraction of human thinking, small changes on them can lead to completely different meanings and thus the semantic gap becomes larger. For instance, advances on technical document understanding has recently focused on room detection and floor-plan understanding [1, 2] or on patent image retrieval since patent retrieval is performed through flowchart diagram recognition [3]. In these tasks, symbol retrieval is useful, but the variability of symbols belonging to the same semantic class comparing the variability between symbols from different semantic classes makes the recognition task more difficult than usual.

A key-step in any retrieval system is the descriptor used to represent the objects of interest. For sketched and handmade symbols the challenge is finding robust descriptors to geometric distortions while remaining discriminative enough [4–7]. Indeed, these distortions are very close to the geometric distortions found in objects in general purpose recognition tasks. This may explain why recently wide-spread descriptors such as SIFT [8] or HoG [9] have been used in handwriting word spotting with various degree of success [10,11]. However, there is still a limitation on the performance of such descriptors when they are applied to multi-writer documents.

The *Bag of Visual Words* (BoW) is one the most used framework in objects recognition applications [12,13], text detection [14], image classification [15,16] and symbol description [17]. In this framework, two issues are addressed: the image, or object, description and the visual vocabulary construction. Concerning the description, SIFT and HoG descriptors are among the two kind of feature vectors commonly used in computer vision applications, while Shape Context [18] has been applied in symbol recognition applications [17]. Concerning visual vocabulary generation, K-means and product quantization [19] are two popular algorithms used for this purpose. In any case, each feature vector is assigned to a cluster centroid, which plays the role of *visual word*. Indeed, product quantization is the sparsest representation we can assign to a feature vector. In fact, product quantization can sometimes be even too restrictive, giving rise to a coarse description of the feature vector [15]. Thus, sparse representation methods, a.k.a *sparse coding* methods, allow to represent feature vectors by a reduced number of linear combination of visual words [20, 21]. By sparse representation techniques we refer to the collection of optimization methods seeking the minimum number of visual words needed to represent a given feature vector. Sparse representation methods have already been applied in other related tasks such as human action recognition [22, 23], face recognition [24–26] or image classification [27]. In particular, a visual dictionary was learned in [25] from the Radon transform of labeled images of faces. In that work, the use of the Radon transform allowed the learning of a visual dictionary invariant under a subset of linear transforms.

The architecture of a BoW approach with sparse representations is composed of three modules: feature vector construction, visual vocabulary generation and sparse representation. The feature vector construction module is similar to the feature vector construction module in any BoW approach. The visual vocabulary is obtained after applying the K-SVD algorithm, which can be seen as a generalization of the K-means algorithm [28]. Finally, sparse representations are achieved after applying any optimization algorithm [29–32].

In this paper we study how to use sparse representations for symbol description in retrieval tasks. To the best of our knowledge, this is the first attempt of using this kind of representation in symbol retrieval tasks. Unlike classifications tasks, we need to find how to link a sparsest symbol representation with a retrieval purpose. To take advantage of these representations we need to solve two issues. The first one concerns the visual vocabulary construction. We have solved it by means of the K-SVD algorithm using the SCIP descriptor. The second issue concerns the retrieval phase. We propose the sparse vector model by extending the *tf-idf* model [33] to sparse representations.

The rest of this paper is organized as follows. In Section 2, we briefly overview some of the relevant shape descriptors used in document analysis as well as some fundamental backgrounds on the shape context and the shape context of interest points. Then, in section 3 we recall the main properties of sparse representation methods. Next, in section 4 we review the K-SVD algorithm and we explain how it is applied to symbol descriptors. In section 5, we extend the vector model to sparse representations for symbol retrieval tasks. To conclude, we report experimental results in section 6 and we discuss the results and future works in section 7.

2 Overview of Symbol Descriptors

There is a large literature on symbol recognition since it is a required step in many computer vision tasks. Although each field has developed their own specialized descriptors some of them has proved to work reasonably well in a wide range of applications. This is the case of local descriptors such as SIFT [8], HoG [9] or SC [18], among others. In this section, we will review those descriptors used in the field of document analysis. We have followed the taxonomy used in [34] for symbol recognition methods where symbol recognition techniques are analyzed from two different point of views: *description* and *recognition*. Symbol description focuses on the definition of either local or global shape descriptors; invariant to similarity transforms; and robust to local symbol distortions and document noise. Conversely, symbol recognition focuses on the methods used to perform the recognition task. Since we are interested on the properties of symbol descriptors in this section we will summarize them from this point of view. Moreover, we will devote special focus to Shape Context (SC) descriptor and its local extension to interest points: the Shape Context of Interest Points (SCIP) descriptor [17].

Symbol descriptors can be divided into different groups depending on the properties of *primitives* used to be computed, the *feature extraction method* applied, and the data representation used for each *descriptor* [34]. Descriptors based on pixel primitives like moments [4, 35, 36], generic Fourier descriptors (GFD) [37] and SC [18] are invariant to translation, scaling, and rotation. They provide a global description of the whole symbol and consequently, it is assumed that symbols have been correctly segmented. However, in technical documents, symbols are non-isolated and they are affected by partial occlusions. In such case, the performance of these descriptors significantly decays since they are based on the inner shape of the symbol. On the contrary, contour-based and skeleton-based descriptors seem to be more robust to partial occlusions than pixel-based descriptors, since their performance usually decrease less than the performance of pixel-based descriptors [34, 38].

There are also primitives encoding geometric information [39, 40]. In general, these descriptors are invariant under similarity transformations, but it depends on a prior normalization step, which is very sensitive to noise. Although these descriptors can easily be computed, either they are usually poorly discriminant [41] or the matching process is time consuming [42].

Syntactic and structural descriptors are suitable for symbol description since differences between symbols come from differences between spatial relation between primitives (e.g. lines, arcs). Some examples of such descriptors are rule-based [6, 43], strings [44] and attribute relational graph (ARG) [45] but their performance are highly affected by noisy data. In general, the time complexity of matching algorithms of *structural* descriptors is still an important drawback in many symbol retrieval systems, although some attempts to speed up the process have been proposed [46].

In summary, global descriptors are hard to apply in documents with non-isolated symbols while the time complexity of syntactic and structural methods make their use on retrieval tasks still challenging. Thus, local descriptors like SIFT [47] and the variant of the SC introduced in [17] are more suitable for symbol retrieval tasks. The remainder of this section is devoted to review SC and its variant the SCIP descriptor.

2.1 Shape Context

The *Shape context* is one of the descriptors with higher accuracy rates in many shapes recognition tasks [18, 48, 49]. Shape boundaries, either internal or external, are sampled in n points. For each point p_i on the symbol contour, its coarse histogram h_i of the relative coordinates of the remaining $n-1$ points is computed as: $h_i(l) = \#\{c \neq p_i | (c - p_i) \in \text{bin}(l)\}$, where $l = 1, \ldots, L$ and 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 shape context is a real matrix $H = \{h_1, \ldots, h_n\}$ with dimensions $L \times n$, see Fig. 1. Since histograms h_i are computed with respect to all sampled points from shape contours, SC is invariant under translation. Scale invariance is obtained by dividing all radial distances by the mean distance among all pair of points. Moreover, it is inherently insensitive to small perturbations of symbol contours and therefore, it is robust to small nonlinear transforms.

2.2 Shape Context of Interest Points

The SC descriptor has two main drawbacks highlighted at the beginning of this section when it is applied to symbol retrieval tasks. It is a global descriptor and the matching function is computationally time-demanding if the number of boundary points is large.

Inspired by the works of object detection based on key-points detection [8,50], SC was extended to be computed on interest points [17]. In that approach, called *Shape context of interest point* (SCIP), a symbol is described by a set of local SC descriptors, each of them computed on interest points. Given a symbol, the interest points IP and the contour points C are detected. Each interest point is represented by its coordinates and the dominant orientation: $p_i = \{x_i, y_i, \vec{e_i}\}$, while the relative log-polar coordinates of contour points $c_j \in C$ is denoted by $c_{ij} = (\log r_{ij}, \theta_{ij})$. The normalized distance from p_i to c_j is r_{ij} and $\theta_{ij} = \langle \overrightarrow{p_i c_j}, \vec{e_i} \rangle$. Then, the histogram at p_i is computed as: $h_i(l) = \#\{c_{ij} \neq p_i | (c_{ij} - p_i) \in \text{bin}(l)\}$, $l = 1, \ldots, L$. Rotation invariance cannot obtained as for the SC descriptor because the set of interest points is rarely a subset of contour points in most of the cases [51]. Instead, the dominant orientation of interest points θ_{ij} was used to obtain rotation invariance.

3 Sparse Representation

The sparse representation of a signal h is a linear combination of a few elements of a given dictionary. More precisely, the sparse representation is the solution of the under-determined linear system of equations $h = Dx$ for a given dictionary $D = \{d_1, d_2, \ldots, d_K\} \in \mathbb{R}^{L \times K}$ and input signal $h \in \mathbb{R}^L$, $x \in \mathbb{R}^K$, with $K \gg L$. If D is a full-rank matrix, there are an infinity of solutions x_i satisfying such linear system. Thus, the sparse representation x is the solution of the last linear system having the smallest number of non-zero coefficients. This idea can be expressed through the definition of a constrained optimization problem subject to the linear constraint $Dx = h$. The objective function $f(x)$ measuring the sparsity of x is:

$$\min_x f(x) \text{ subject to } Dx = h \tag{P_f}$$

If $f(x)$ is the l_0 pseudo-norm $\|x\|_0$ (number nonzero elements in vector x), then the problem (P_f) becomes finding the sparse representation x of h satisfying:

$$\min_x \|x\|_0 \text{ subject to } Dx = h \tag{P_0}$$

Finding the exact solutions of (P_0) is a NP-hard problem [52]. Therefore, research has been done on algorithms to find approximate solutions. For instance, greedy algorithms are iterative algorithms performing local optimization with the expectation of converging to a global optimum. Examples of such kind of algorithms applied to sparse representation are Matching Pursuit (MP) [53], Orthogonal-MP (OMP) [29], Weak-MP [54], the Thresholding algorithm [55], and other [56].

Other approaches replace the l_0-norm by other functions $f(x)$ to convert the NP-hard problem (P_f) to a constrained optimization problem solved in polynomial time. Some examples of these surrogate objective functions are l_p-norms, for $p \in (0,1]$, and smooth functions such as $\sum_i \log(1+\alpha x_i^2)$, $\sum_i x_i^2/(\alpha+x_i^2)$ or $\sum_i(1-exp(-\alpha x_i^2))$. The FOcal Underdetermined System Solver (FOCUSS) [57] and the basis pursuit (BP) methods [30,31,58] are some examples of methods belonging to this family of algorithms. In FOCUSS, the objective function is a weighted l_2-norm and the sparse representation x is found thanks to the Iterative-Reweighed-Least-Squares (IRLS) algorithm [58]. On the contrary, the l_1-norm is the objective function used on BP methods and the (P_0) problem was redefined as:

$$\min_x \|W^{-1}x\|_1 \text{ subject to } Dx = h \qquad (P_1^W)$$

where W is a diagonal positive-definite matrix [31]. A natural choice for each entry in W is $W_{i,i} = 1/\|a_i\|_2$. If $\tilde{x} = W^{-1}x$, then the problem (P_1^W) is:

$$\min_{\tilde{x}} \|\tilde{x}\|_1 \text{ subject to } h = DW\tilde{x} = \tilde{D}\tilde{x} \qquad (P_1)$$

and $\tilde{D}$ is the normalized version of D and the sparse representation x is computed from $\tilde{x}$. This is the classic definition of the BP method [59]. (P_1) is usually solved with a normalized dictionary matrix D using linear programming or IRLS methods, [30, 58]. The BP algorithm is computationally more intense than the MP, but it achieves sparser representations. Moreover, under appropriate conditions on D and x, the BP and the OMP algorithms give the unique solution of (P_1) and (P_0) [31].

The linear constraint $Dx = h$ used in the above constrained optimization problems become too restrictive in many real applications with noisy data. Assuming that noise has finite energy: $\|e\|_2^2 \leq \epsilon^2$; the signal h is modeled as the sum of the linear combination of sparse vector x and noise e: $h = Dx + e$.

Then, the exact constraint $h = Dx$ is relaxed by $\|Dx - h\|_2 \leq \epsilon$, being $\epsilon \geq 0$ the tolerance error. Finally, the sparsest solution x has to satisfy:

$$\min_x \|x\|_0 \text{ subject to } \|Dx - h\|_2 \leq \epsilon \tag{P_0^ϵ}$$

Similarly, to the non-noisy solvers, the l_0-norm in (P_0^ϵ) can be replaced by other l_p-norms, such as for instance l_1, l_2, or l_∞. In particular, the basis pursuit denoising (BPDN) problem defined below is obtained when the objective function is replaced by the l_1-norm:

$$\min_x \|x\|_1 \text{ subject to } \|Dx - h\|_2 \leq \epsilon \tag{P_1^ϵ}$$

Lagrange multipliers were used in [55] to solve (P_1^ϵ) following:

$$\min_x \lambda\|x\|_1 + \frac{1}{2}\|Dx - h\|_2^2 \tag{Q_1^λ}$$

There is also a huge literature of optimization methods tackling the problem (Q_1^λ). On the one hand, linear regression techniques like the Least Absolute Shrinkage and Selection Operator (LASSO) method [60] and the Least Angle Regression Stagewise (LARS) method [61] can be applied if (Q_1^λ) is considered as a regularized regression problem. On the other hand, the minimization of the objective function in (Q_1^λ) in its more general form can be treated using various classic iterative optimization algorithms, such as for instance the Steepest-Descent algorithm, the Conjugate-Gradient algorithm or the interior-point algorithm. However, in the case of high-dimensional problem, these methods perform very poorly and the Iterative-Shrinkage algorithms has been developed. Some algorithms in this last family of algorithms include the Stagewise Orthogonal-Matching-Pursuit (StOMP) algorithm [62], the EM and the Bound-Optimization approaches [63, 64], the IRLS-based shrinkage algorithm and the Parallel-Coordinate-Descent (PCD) algorithm [55].

4 Dictionary Learning

In the previous section we have reviewed the existing optimization methods for finding the sparsest representation of an input signal given a visual dictionary. This dictionary could be composed of functions, such as for instance Fourier basis, wavelets or curvelets frames, just to enumerate a few. However, these families of functions are not always suitable to sparsely represent complex objects like symbols. Thus, dictionary learning algorithms have been developed, such as the Method of Optimal Directions (MOD) [65] and the K-SVD algorithm [28]. The performance of these two algorithms is similar with a small advantage for the K-SVD [66]. So, we have chosen the K-SVD algorithm for SCIP dictionary learning although other learning algorithms may be used. In this section, we explain how to initialize and how to use the K-SVD to obtain a sparse representation of the SCIP descriptor (section 2.2).

4.1 The K-SVD Algorithm

Let $H = \{h_m\}_{m=1}^{M}$ be the set of M real-value vectors computed from the training images dataset. These vectors will later correspond to SCIP descriptors but for the sake of simplicity, we just assume in this section that they are vectors in $\mathbb{R}^L$. The learned dictionary is the solution of:

$$\min_{D,x_m} \sum_m \|x_m\|_0 \text{ subject to } \|h_m - Dx_m\|_2^2 \leq \epsilon \qquad (1)$$

which is similar to (P_1^ϵ) but now, the dictionary $D \in \mathbb{R}^{L\times K}$ is also unknown and it has to be found during the optimization process. The goal of the K-SVD [28] is to find a visual dictionary D and the sparse representations of descriptors in the training set, see Algorithm 1. More specifically, D is updated at each iteration following two main steps: the *sparse representation* and the *update dictionary* steps. In the sparse representation step, all the sparse representations $\{x_m\}$ are computed while keeping D fixed. These sparse representations can be computed by any algorithm solving (P_1^ϵ). We have used the OMP algorithm, as in [28]. In the *update dictionary* step, each element of the dictionary (a column in matrix D) is sequentially updated.

The residual error used as stopping condition in the K-SVD algorithm is defined in (2). $x_{k_0}^T \in \mathbb{R}^M$ is the k_0-th row in X and the notation $\|\cdot\|_F$ stands for the Frobenius norm. Thereby, the residual error is minimized for each visual word d_{k_0} while keeping fixed the sparse representation X, found in the previous stage, and the other visual words.

$$\begin{aligned}\|H - DX\|_F^2 = \|H - \sum_{k=1}^{K} d_k x_k^T\|_F^2 &= \|(H - \sum_{k \neq k_0} d_k x_k^T) - d_{k_0} x_{k_0}^T\|_F^2 \\ &= \|E_{k_0} - d_{k_0} x_{k_0}^T\|_F^2\end{aligned} \tag{2}$$

Then, the singular value decomposition (SVD) of error matrices $E_{k_0} = H - \sum_{k \neq k_0} d_k x_k^T = U_{k_0} S_{k_0} V_{k_0}$ is proposed in order to reduce the approximation error [28]. Both, the k_0-th visual word d_{k_0} and $x_{k_0}^T$, are respectively replaced by the first eigenvector of U_{k_0} and by the product of the first eigenvector of V_{k_0} and the first diagonal element of S_{K_0}. However, the new vector $x_{k_0}^T$ is very likely to be non-sparse. To overcome this problem, a matrix Ω_{k_0} with convenient dimensions is defined to restrict E_{k_0} and $x_{k_0}^T$ only on those columns of X where the entry is non-zero. Thus, the approximation error is:

$$\|E_{k_0}\Omega_{k_0} - d_{k_0} x_{k_0}^T \Omega_{k_0}\|_F^2 = \|E_{k_0}^R - d_{k_0} x_{k_0}^R\|_F^2 \tag{3}$$

and the SVD is applied to the restricted version of error matrices: $E_{j_0}^R$. Thus, the sparsity constraint is not violated after applying the SVD.

4.2 SCIP Sparse Dictionary

In the context of symbol retrieval, the training set is composed of N instances of symbols images and, for each of them, we have r_n SCIP descriptors: $H^{(n)} = \{h_1^{(n)}, h_2^{(n)}, \ldots, h_{r_n}^{(n)}\}$. $H = \bigcup H^{(n)}$ is the whole set of all SCIP descriptors whatever the symbol and it is the set of SCIP descriptors used as training set in the K-SVD as depicted in Algorithm 1.

The dictionary matrix D is initialized by randomly choosing K descriptors from the training set H and then normalized column-wise by the l_2-norm. The K-SVD is run until a maximum number of iterations is reached or an approximation error is smaller than a fixed ϵ. After applying the K-SVD algorithm, we have learned a visual dictionary $D \in \mathbb{R}^{L \times K}$, which provides the sparse representations of all the SCIP descriptors in H. Each column of a visual dictionary D will be a visual word.

5 Sparse Vector Model

So far, we have briefly reviewed the main family of shape descriptors used in symbol retrieval, the main algorithms used to find sparse representations and we have also explained one of the main algorithms for learning overcomplete dictionaries needed in sparse representation methods. In this section, we explain how to combine sparse representation of SCIP descriptors with the vector model framework [33] used in retrieval tasks.

In general, vector model provides a representation of the more discriminative words at document level. Thus, a K dimensional vector of real values denote the relative importance of words according to the number of occurrences of such word in the document (*tf* factor) and its discriminative capacity (*idf* factor). Similarly, we propose a sparse vector model at symbol image level.

Without loss of generality, we can assume that $h \in \mathbb{R}^L$ is one of the SCIP descriptor in H and $x \in \mathbb{R}^K$ is the sparse representation of h given the dictionary D. Instead of using vector quantization techniques and assigning a single visual word to each SCIP descriptor [12, 17], we can see h as a linear combination of visual words.

We denote by v_i^n the characteristic vector of h_i^n. $v_i^n \in \mathbb{R}^K$ is the 0-1 vector obtained from the sparse representation x_i^n given D. More specifically, let $x_i^n = (\alpha_1, \ldots, \alpha_p, \ldots, \alpha_K)$ be the sparse representation of h_i^n in D. The reconstructed descriptor of h_i^n, denoted with $\bar{h}_i^n$, is computed by:

$$\bar{h}_i^n = \sum_{j=1}^{K} \alpha_j \cdot d_j \tag{4}$$

where d_j is the j-th visual word of dictionary D. Moreover, the number of zero elements in x_i^n is larger than the number of nonzero elements because x_i^n is sparse. Let I be the indexes set where x_i^n is different to 0, then we define the characteristic vector $v_i^n(k) = 1$ if $k \in I$ and 0 otherwise. For example, if $x_i^n = (\alpha_1, 0_2, \ldots, \alpha_{p-1}, \alpha_p, 0_{p+1}, \ldots, \alpha_q, \alpha_{q+1}, 0_{q+2}, \ldots, 0_K)$ then $I = \{1, p-1, p, q, q+1\}$ and $v_i^n = (1_1, 0_2, \ldots, 1_{p-1}, 1_p, 0_{p+1}, \ldots, 1_q, 1_{q+1}, 0_{q+2}, \ldots, 0_K)$. Thus, v_i^n is a vector where the 1 coefficients indicate the presence of a visual word. Next, we define *tf* and *idf* factors to describe, respectively, document contents and the importance degree of terms as follows: f_k^n is the frequency of the word k that appears in the symbol n and $tf_{k,n} = \frac{f_k^n}{\max_s f_s^n}$. Observe that we can easily compute these frequencies through the characteristic vector: $f_k^n = \sum_i^{r_n} v_i^n(k)$.

The *idf* factor is defined as usual in information retrieval systems but also adapting its definition to the sparse representation of SCIP descriptors. The importance in distinguishing a relevant symbol from non-relevant one in the database is measured by $\log \frac{N}{l_k}$, where l_k is the number of symbols in which the word k appears: $l_k = \#\{n = 1, \dots, N | f_k^n \neq 0\}$. Therefore, the vector model w^n for a given symbol is defined by the set of weighted frequencies:

$$w_k^n = tf_{k,n} \times idf_k = \frac{f_k^n}{\max_s f_s^n} \times \log \frac{N}{l_k} \tag{5}$$

5.1 Symbol Retrieval

We perform symbol retrieval using our sparse vector model. Thus, for each query symbol s^q, its sparse vector model is computed as explained previously. The similarity between the query symbol s^q and symbols s^n in the database is computed as the cosine distance between the two vectors w^q and w^n:

$$\text{distance}(w^q, w^n) = \frac{\langle w^q, w^n \rangle}{|w^q| \times |w^n|} \tag{6}$$

where $\langle \cdot, \cdot \rangle$ is the dot product. Finally, the retrieved symbols from the database are ranked based on their similarity to the query symbol s^q.

6 Experimental Results

The experiments designed in this section are devoted to show that the proposed scheme outperforms other related approaches used in the literature to recognize symbols. Moreover, one of the main challenge in our method is its capacity to retrieve *good* symbol even if the queried symbol does not appear in the learning dataset.

We have designed three kinds of experiments that have allowed us to judge the performance of the sparse vector model approach under several conditions of the training set. We have devoted the first group of experiments in finding the best parameters of the symbol retrieval system following benchmark datasets. The second group of experiments, aimed at evaluating the performance of the SCIP descriptor comparing to other shape descriptors. In addition, we have compared the performance of K-SVD with the K-means algorithm for visual vocabulary generation. We have analyzed the results with different degrees of geometric distortions, noise, rotation and scale transformations. The last set of experiments seeks to evaluate the retrieval system in a more realistic configuration in which some symbol images does not exist in the training set.

6.1 Datasets and Performance Evaluation

We have considered three public datasets. The first dataset is the synthetic GREC2003 dataset, used in many symbol recognition contests since 2003[1]. The second dataset, called herein CVC dataset, is a handwritten version of the GREC2003 dataset, and it was created at the Barcelona Computer Vision Center [6]. The third dataset is the FRESH dataset composed of real symbol instances [67].

- The GREC2003 dataset [68] is composed of 520 symbols. It was created to evaluate the performance of symbol descriptors under different geometric distortions of symbols, rigid transforms (rotation and scale) and noise simulating document degradation. More precisely, different images were created depending on degradation levels, geometrical distortions and rigid trans-

[1] http://www.cvc.uab.es/grec2003/SymRecContest/

formations applied on them. We have taken the images of such tests and grouped them into three different subsets: the first one is composed of only rotated and scaled symbols (300 images) while the second one is composed of geometrically distorted instances (115) of original symbols. The last subset is composed of distorted and noisy instances of symbols (105).

- The CVC dataset of handwritten symbols is composed of 502 handwritten symbols drawn by 14 different volunteers people from the CVC. These volunteers have taken the same symbol classes of the GREC2003 dataset .
- The FRESH dataset is composed of 144 segmented symbols extracted from aircraft electrical wiring diagrams of real world industrial drawings. Differences between symbols are due to slight details in symbols. In addition, the number of instances of each class of symbol is imbalanced having only a few for some of them. Consequently, symbols were grouped into *similar* symbol semantic classes according the agreement of 6 volunteers which had participated in the ground-truth generation [67].

For evaluation purposes, we have created two partitions for each dataset: training and test sets. These partitions have randomly generated and consequently, instances of the same class symbol can appear in both partitions; except for the experiments described in section 6.4, where symbol classes in the test partition does not appear in the training partition. The training partition has been used for learning the descriptors dictionary. The test set is composed of the symbol images used as queries in the evaluation. The size of the training and test sets is respectively around 85% and 15%.

We have used the implementation introduced in [69] to learn the visual dictionary of SCIP descriptors. We have computed Precision and Recall curves and we have reported the obtained results using the *area under the precision-recall* *curve* (AUC-PR) metric [70] as a simple metric to evaluate the performance of the proposed scheme and compare it against related methods. We have applied bootstrapping for performance evaluation and we have repeated 10 times each experiment due to the random initialization of the K-SVD algorithm. We have performed a two-sided Wilcoxon rank sum test with 5% of significance level to assess if differences on the AUC-PR values are significant, or not, from a statistical viewpoint. The two-sided Wilcoxon rank sum test is a non-parametric hypothesis test used to compare the means of two continuous distributions. This test is similar to the usual t-student test but the normality assumption is not needed [71].

6.2 Study of Parameters

This set of experiments aims at finding the best parameters for the proposed scheme: the dimension of SCIP descriptor, the approximation error and the size of the visual dictionary, respectively named L, ϵ and K. An exhaustive search in the parameter space will require a run of the K-SVD algorithm for each combination of these three parameters.

Since the learned dictionary A is an over-complete dictionary, the number of columns (K) needs to be larger than the number of rows (L). In addition, a large K implies sparser descriptor representations. However, if the size of K is too large, it leads to high computing time issue. In fact, we have experimentally found that a size of $K = 512$ is a good trade-off between the performance and the computation time. Therefore, we have fixed $K = 512$ for all the experiments done in this paper.

Concerning the SCIP dimension, we have sampled the radial and angular parameters respectively in the following set of values: $\{3, 4, 5\}$ and $\{12, 16\}$ given the following dimensions: $L = \{36, 48, 60, 56, 80\}$. Furthermore, we have sampled the approximation error ϵ, defined in problem (P_1^ϵ), in a interval of values ranging from 0.01 to 0.2.

We have learned a dictionary on the training subset corresponding to the three datasets (GREC 2003, CVC and FRESH) for each pair (L, ϵ) of the parameter space. For each training dataset, we have computed the AUC-PR values and we have repeated this scheme 10 times since the K-SVD is randomly initialized. Tables 1-3 show the average of AUC-PR values for each dataset over the 10 experiment repetitions. In these tables, the best values are in bold and an entry marked by (-) indicates that the corresponding pair of parameters performs worst than the best set of parameters. An entry marked by (=) indicates that the obtained results are not significantly different. In those cases where there is not significant difference between the compared parameters will simply mean that the choice of L and ϵ has not a real impact on the performance of the retrieval system.

To summarize these experiments, in Table 4 we have reported the AUC-PR values and the average time required to process a query symbol for the best parameters. The (+) on the right side of the average of AUC-PR values indicates that the AUC-PR values obtained for the best configuration are significantly better comparing to other pairs (L, ϵ). Moreover, note that the average time per query in the FRESH dataset is approximatively 20s while for the other datasets it takes less than 2 seconds. The main reason is that the size of the images in this dataset is much larger than the other (around 4 times compared to GREC dataset and 16 times for CVC dataset).

6.3 Invariance and Robustness

The experiments reported in this section aim at evaluating whether sparse representation keeps invariance properties regarding affine transforms but also whether it improves the overall performance of symbol retrieval systems. In other words, our working hypothesis is that if a descriptor is invariant to any affine transform, then its sparse representation is also invariant to the same transforms and distortions. To evaluate the performance of sparse vector model of SCIP descriptors we have compared it to 6 state-of-the-art descriptors (reviewed in section 2), namely R-signature [72], GFD [73], Zernike moments [74], SIFT [8], SC [18] and SCIP [17]. In addition, we have used two descriptors for Zernike moments. The first descriptor, G_1 includes 32 low-order moments while the second descriptor, G_2, includes 32 high-order moments. We have only considered the magnitude of Zernike moments for both descriptors G_1 and G_2. These descriptors have been used frequently and successfully in the literature for general symbol recognition purposes. For instance, Radon transform [75], Shape Context [17], GFD [73] and Zernike moment [76,77] for shape recognition and retrieval; SIFT has been applied for trademark and logo detection [78,79]. All these descriptors have been applied on the three benchmark datasets. In addition, the non-sparse version of SCIP and SIFT descriptors also depends on the random initialization of the k-means algorithm. Consequently, we have also repeated the experiments with these descriptors 10 times, as we have done for their sparse representation. Finally, we have used the learned dictionaries in the previous experiment and their respective best values for L and ϵ. To avoid the effect of the random initialization we have used again the same 10 dictionaries learned in the previous experiment. We can observe from

Table 5 that there is a significant improvement of symbol retrieval schemes using SCIP descriptors compared to Zernike moments, SC and R-signature descriptor. Only the GFD achieves similar results than the proposed scheme in the degraded set of the GREC2003 dataset. This result can be explained by two facts. On the one hand, we have applied GFD to the whole image, since symbols in this dataset are fully segmented. On the other hand, the key-points detection step is sensitive to noise. This fact explains the poor performance of SIFT descriptor on most of the datasets also. As pointed out in [17], the SIFT descriptor looses its effectiveness when working with symbols.

Nevertheless, we can remark that constructing a vector model from the sparse representations of descriptors provides better results than using cluster algorithms like k-means. We conclude this experiment by giving some examples of the proposed method. Figure 3 and 4 show examples of the nearest retrieved symbols for some queries from the three datasets (GREC,CVC and FRESH). We can see there that the retrieved symbols are almost the same regardless the rotation, the scale, small distortions and deformations.

6.4 Unseen symbols

One of the main difficulties in symbol recognition and any symbol retrieval system is the relative few number of instances of each kind of symbol. This fact makes the task of any learning algorithm harder and consequently their performance usually drop down. The sparse representation proposed in this work is based on the learning of a dictionary of SCIP descriptors that have been computed on a training dataset composed of a given set of kind of symbols. It seems quite obvious that symbols in the training dataset have an impact in the final sparse representation. Since the dictionary is learned on a descriptor which is invariant to scale, rotation and robust to small perturbations, the question is whether the learned dictionary has the capacity of describing symbols that have not been learned, *unseen symbols*. The purpose of this experiment is to evaluate the discrimination capacity of the proposed representation to retrieve unseen symbol instances.

We have defined this experiment as follows. For each dataset, we have considered different subsets of the training dataset, namely 25%, 50% and 75% and we have compared the performance to the retrieval system using the whole training dataset. Indeed, symbols used for training sets have been randomly selected, and consequently we have repeated again 10 times all the experiments. The parameters used for building the dictionary of this subsets are the best values of L and ϵ found in Section (6.2) and as usual the two-sided Wilcoxon sum rank test is performed to asses if observed differences are statistically significant or not. Finally, we have performed this experiment for SCIP and SIFT descriptors to see if the performance behavior depends on chosen descriptor. We can see from Table 6 that, for the GREC2003 and the FRESH dataset, there are small significant differences when using a smaller training compared to the whole dataset. In most cases, one quarter of the learning set provides the same results as using the whole data. The performance for the GREC dataset, for the 50% partition, decreases a bit for both SCIP and SIFT descriptors although it is not significantly different. We explain this behavior by the random process followed to generate the partition sets. Only the retrieval performance, in the 25% subsets and for the SIFT descriptor, seems to decrease a bit. Nevertheless, the behavior of SCIP and SIFT descriptor is quite similar in all cases. To conclude this experiment, we can say that one advantage of our sparse representation is its capacity to generalize the representation to unseen symbols.

7 Conclusions and Future Work

In this paper we have applied a sparse representation to visual vocabulary for symbol recognition tasks. Thus, we have taken a symbol descriptor and we have adapted the sparse representation theory as well as the learning dictionary algorithm. Moreover, we have extended the traditional vector model used in retrieval tasks, to sparse representation. The proposed approach is general and easily applied to any descriptor. We have evaluated this approach on several reference symbol datasets and the reported results show a stable behavior of the system. It means that, although a parameter tuning has to be done in order to select the best system parameter, there is a wide range of values shared by all the datasets and given similar results. Moreover, we have studied the robustness of sparse representation to affine transforms and symbols distortions. The reported results show a good behavior of our approach compared to related state-of-art methods and to generalize the representation for symbols which have been affected by small perturbations or affine transformations. Finally, we have seen the capacity of sparse descriptors to represent unseen symbols.

Nevertheless there are still open issues in which we are currently working on. First of all, we are extending this work to symbol spotting tasks in large technical documents where symbols cannot be easily well segmented. Finally, we have to see how to apply on-line learning methods to progressively enrich our visual dictionary when new symbol instances appear.

Acknowledgements

This work was partially supported by the European SCANPLAN (A0806017L), the Spanish Co-READ (TIN2012-37475-C02-02) and the 2014-SGR-1436 projects.

References

[1] R. Wessel, I. Blümel, R. Klein, The room connectivity graph: Shape retrieval in the architectural domain, in: V. Skala (Ed.), The 16h International Conference on Computer Graphics, Visualization and Computer Vision, 2008.

[2] M. Weber, M. Liwicki, A. Dengel, a.scatch - a sketch-based retrieval for architectural floor plans, in: ICFHR, 2010, pp. 289–294.

[3] M. Rusinol, L.-P. de las Heras, O. Ramos Terrades, Flowchart recognition for non-textual information retrieval in patent search, Information Retrieval 17 (5-6) (2014) 545–562.

[4] T. Cheng, J. Khan, H. Liu, D. Yun, A symbol recognition system, in: ICDAR, 1993, pp. 918–921.

[5] D. Zuwala, S. Tabbone, A method for symbol spotting in graphical documents, in: Int. Workshop on DAS, Vol. 3872, 2006, pp. 518–528.

[6] J. Mas, J. Jorge, G. Sánchez, J. Lladós, Representing and parsing sketched symbols using adjacency grammars and a grid-directed parser, Vol. 5046, Graphics Recognition. Recent Advances and New Opportunities, 2008, pp. 169–180.

[7] T. Y. Ouyang, R. Davis, A visual approach to sketched symbol recognition, in: 21st international jont conference on Artifical intelligence, 2009, pp. 1463–1468.

[8] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[9] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Computer Vision and Pattern Recognition, Vol. 1, 2005, pp. 886–893.

[10] M. Rusinol, D. Aldavert, R. Toledo, J. Lladós, Browsing heterogeneous document collections by a segmentation-free word spotting method, in: Proceedings of ICDAR, 2011, pp. 63–67.

[11] J. Almazan, A. Gordo, A. Fornes, E. Valveny, Efficient exemplar word spotting, in: 23rd BMVC, 2012, pp. 67.1–67.11.

[12] J. Sivic, A. Zisserman, Video google: a text retrieval approach to object matching in videos, in: ICCV, 2003, pp. 1470 –1477.

[13] Q. Li, H. Zhang, J. Guo, B. Bhanu, L. An, Reference-based scheme combined with k-svd for scene image categorization, Signal Processing Letters 20 (1) (2013) 67–70.

[14] M. Zhao, S. Li, J. Kwok, Text detection in images using sparse representation with discriminative dictionaries, Image and Vision Comp. 28 (2010) 1590–1599.

[15] J. Yang, K. Yu, Y. Gong, T. S. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: CVPR, IEEE, 2009, pp. 1794–1801.

[16] Z. Jiang, Z. Lin, L. S. Davis, Learning a discriminative dictionary for sparse coding via label consistent k-svd, in: CVPR, IEEE, 2011, pp. 1697–1704.

[17] T. O. Nguyen, S. Tabbone, O. R. Terrades, Symbol descriptor based on shape context and vector model of information retrieval, in: Proceedings of DAS, 2008, pp. 191 –197.

[18] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (24) (2002) 509–522.

[19] H. Jégou, M. Douze, C. Schmid, Product quantization for nearest neighbor search, IEEE Trans. Pattern Anal. Mach. Intell. 33 (1) (2011) 117–128.

[20] H. Cheng, Z. Liu, L. Yang, X. Chen, Sparse representation and learning in visual recognition: Theory and applications, Signal Process. 93 (6) (2013) 1408–1425.

[21] T. Ge, Q. Ke, J. Sun, Sparse-coded features for image retrieval, in: British

machine vision conference, 2013.

[22] Q. Qiu, Z. Jiang, R. Chellappa, Sparse dictionary-based representation and recognition of action attributes, in: Proceedings of ICCV, 2011, pp. 707–714.

[23] Z. Lu, Y. Peng, Latent semantic learning with structured sparse representation for human action recognition, Pattern Recognition 46 (7) (2013) 1799 – 1809.

[24] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.

[25] Y.-C. Chen, C. S. Sastry, V. M. Patel, P. J. Philipps, R. Chellappa, In-plane rotation and scale invariant clustering using dictionaries, IEEE Trans Image Process 22 (6) (2013) 2166–2180.

[26] H. Zhang, Y. Zhang, T. S. Huang, Pose-robust face recognition via sparse representation, Pattern Recognition 46 (5) (2013) 1511 – 1521.

[27] S. A. Ptucha R, Lge-ksvd: Robust sparse representation classification, IEEE Trans Image Process 23 (4) (2014) 1737–1750.

[28] M. Aharon, M. Elad, A. M. Bruckstein, K-svd: An algorithm for designing overcomplete dictionaries for sparse representation, IEEE Trans. on Signal Process. 54 (11) (2006) 4311–4322.

[29] Y. Pati, R. Rezaiifar, P. Krishnaprasad, Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition, in: Annual Asilomar Conf. on Signals, Systems, and Comp., 1993, pp. 40–44.

[30] S. Chen, D. Donoho, M. Saunders, Atomic decomposition by basis pursuit, SIAM Journal on Scientific Computing 20 (1) (1998) 33–61.

[31] D. Donoho, M. Elad, Optimally sparse representation in general (nonorthogonal) dictionaries via l^1 minimization, PNAS 100 (5).

[32] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, Journal of the Royal Statistical Society, Series B 67 (2005) 301–320.

[33] G. Salton, A. Wong, C. S. Yang, A vector space model for automatic indexing,

Commun. ACM 18 (11) (1975) 613–620.

[34] S. Tabbone, O. Ramos Terrades, An overview of symbol recognition, in: D. Doermann, K. Tombre (Eds.), Handbook of Document Image Processing and Recognition, Springer London, 2014, pp. 523–551.

[35] A. Khotanzad, Y. Hong, Invariant image recognition by zernike moments, IEEE Trans. Pattern Anal. Mach. Intell. 12 (5) (1990) 489–497.

[36] C. Chong, P. Raveendran, R. Mukundan, Translation and scale invariants of legendre moments, Pattern Recognition 37 (1) (2004) 119–129.

[37] D. Zhang, G. Lu, Shape-based image retrieval using generic fourier descriptor., Signal Process. 17 (2002) 825–848.

[38] E. Valveny, S. Tabbone, E. Philippot, Performance characterization of shape descriptors for symbol representation, in: GREC, Vol. 5046 of LNCS, 2008.

[39] G. Lu, A. Sajjanhar, Region-based shape representation and similarity measure suitable for content-based image retrieval, Multimedia Systems 7 (2) (1999) 165–174.

[40] P. Dosch, J. Lladós, Vectorial signatures for symbol discrimination., in: Graphics Recognition: Recent Advances and Perspectives, Vol. 3088, 2004, pp. 154–165.

[41] J. Neumann, H. Samet, A. Soffer, Integration of local and global shape analysis for logo classification, Pattern Recognition Letters 23 (12) (2002) 1449–1457.

[42] F. Mokhtarian, S. Abbasi, J. Kittler, Robust and efficient shape indexing through curvature scale space, in: Proceedings of the BMVC, 1996, pp. 53–62.

[43] H. Bunke, Attributed programmed graph grammars and their application to schematic diagram interpretation, IEEE Trans. Pattern Anal. Mach. Intell. 4 (6) (1982) 574–582.

[44] H. Wolfson, On curve matching, IEEE Trans. Pattern Anal. Mach. Intell. 12 (5) (1990) 483–489.

[45] J. Lladós, E. Martí, J. Villanueva, Symbol recognition by error-tolerant subgraph matching between region adjacency graphs., IEEE Trans. Pattern

Anal. Mach. Intell. 23 (10) (2001) 1137–1143.

[46] A. Dutta, J. Lladós, U. Pal, A symbol spotting approach in graphical documents by hashing serialized graphs, Pattern Recognition 46 (3) (2013) 752–768.

[47] D. Lowe, Object recognition from local scale-invariant features., in: Proceedings of ICCV, 1999, pp. 1150–1157.

[48] S. Belongie, G. Mori, J. Malik, Matching with shape contexts, in: H. Krim, J. Yezzi, Anthony (Eds.), Statistics and Analysis of Shapes, Modeling and Simulation in Science, Engineering and Technology, 2006, pp. 81–105.

[49] Z. Liu, H. Shen, G. Feng, D. Hu, Tracking objects using shape context matching, Neurocomputing 83 (2012) 47 – 55.

[50] S. Agarwal, A. Awan, D. Roth, Learning to detect objects in images via a sparse, part-based representation, IEEE Trans. Pattern Anal. Mach. Intell. 26 (11) (2004) 1475–1490.

[51] S. Tabbone, L. Alonso, D. Ziou, Behavior of the laplacian of gaussian extrema, Journal of Mathematical Imaging and Vision 23 (1) (2005) 107–128.

[52] E. Amaldi, V. Kann, On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems, Theoretical Computer Science 209 (1998) 237–260.

[53] S. G. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, Signal Process. 41 (12) (1993) 3397–3415.

[54] V. N. Temlyakov, Weak greedy algorithms, Advances in Computational Mathematics 5 (2000) 173–187.

[55] M. Elad, Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing, Springer, 2010.

[56] Y. S. Yuan XT, Forward basis selection for pursuing sparse representations over a dictionary, IEEE Trans. Pattern Anal. Mach. Intell. 35 (12) (2013) 3025–3036.

[57] I. Gonzalez, B. Rao, Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm, IEEE Trans. on Signal Process. 45 (3)

(1997) 600–616.

[58] I. Daubechies, R. Devore, M. Fornasier, C. Gunturk, Iteratively reweighted least squares minimization for sparse recovery, Communications on Pure and Applied Mathematics 63 (1) (2009) 1–38.

[59] S. Mallat, A wavelet tour of signal processing: The Sparse Way, 3rd Edition, Academic Press, 2009.

[60] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society, Series B 58 (1994) 267–288.

[61] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Annals of Statistics 32 (2004) 407–499.

[62] D. L. Donoho, Y. Tsaig, I. Drori, J. L. Starck, Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit, Tech. rep., Stanford University (2006).

[63] M. A. Figueiredo, R. D. Nowak, An em algorithm for wavelet-based image restoration, IEEE Trans Image Process 12 (8) (2003) 906–916.

[64] M. A. Figueiredo, R. D. Nowak, A bound optimization approach to wavelet-based image deconvolution, in: Proceedings of ICIP, Vol. 2, Genoe, Italy, 2005, pp. 782–785.

[65] K. Engan, S. O. Aase, J. Hakon Husoy, Method of optimal directions for frame design, in: Proceedings of ICASSP, 1999, pp. 2443–2446.

[66] T. H. Do, Sparse representations over learned dictionary for document analysis, Ph.D. thesis, University of Lorraine (2014).

[67] K. Santosh, B. Lamiroy, L. Wendling, Symbol recognition using spatial relations, Pattern Recognition Letters 33 (3) (2012) 331–341.

[68] E. Valveny, P. Dosch, Symbol recognition contest: A synthesis, in: J. Lladós, Y.-B. Kwon (Eds.), GREC, Vol. 3088 of LNCS, Springer, 2003, pp. 368–386.

[69] R. Rubinstein, M. Zibulevsky, M. Elad, Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit, Tech. rep. (Apr. 2008).

[70] J. Davis, M. Goadrich, The relationship between precision-recall and roc curves, in: International conference on Machine learning, 2006, pp. 233–240.

[71] J. D. Gibbons, Nonparametric Statistical Inference., Marcel Dekker, 1985.

[72] S. Tabbone, L. Wendling, J.-P. Salmon, A new shape descriptor defined on the radon transform, CVIU 102 (1) (2006) 42–51.

[73] D. Zhang, Generic fourier descriptor for shape-based image retrieval, in: International Conference on Multimedia and Expo, Vol. 1, 2002, pp. 425–428.

[74] A. Tahmasbi, F. Saki, S. B. Shokouhi, Classification of benign and malignant masses based on zernike moments, Computers in Biology and Medicine 41 (8) (2011) 726–735.

[75] S. Tabbone, L. Wendling, Recognition of symbols in grey level line drawings from an adaptation of the radon transform, in: Proceedings of 17th ICPR, Vol. 2, 2004, pp. 570–573.

[76] W.-Y. Kim, Y.-S. Kim, A region-based shape descriptor using zernike moments, Signal Processing: Image Communication 16 (1-2) (2000) 95–102.

[77] C. Kan, M. D. Srinath, Invariant character recognition with zernike and orthogonal fouriermellin moments, Pattern Recognition 35 (1) (2002) 143–154.

[78] A. D. Bagdanov, L. Ballan, M. Bertini, A. D. Bimbo, Trademark matching and retrieval in sports video databases, in: Proceedings of the Int. Workshop on mult. inf. ret., 2007, pp. 79–86.

[79] V. P. Le, N. Nayef, M. Visani, J. Ogier, C. D. Tran, Document retrieval based on logo spotting using key-point matching, in: 22nd ICPR, 2014, pp. 3056–3061.

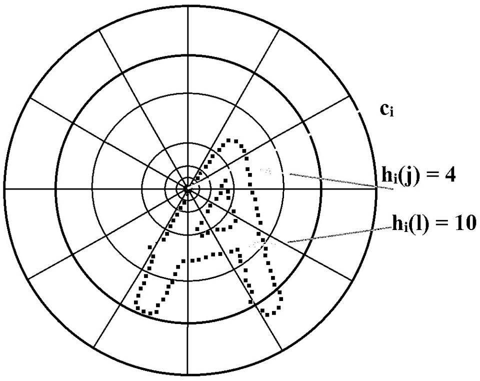


Fig. 1. Illustration about how to compute the shape context

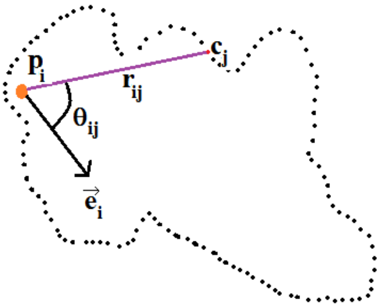


Fig. 2. The relative log-polar coordinates of c_j with regard to p_i

Algorithm 1 Learning algorithm K-SVD

```
INPUT: $A_{(0)} \in R^{L \times K}$; $H$ training set; $t = 0$;
1. Initialize: Normalization the columns of matrix $A_{(0)}$;
2. Main Iteration
while $\|H - A_{(t)}X_{(t)}\|_F^2 > \epsilon$ do
    - Find all sparse representations $X_{(t)}$ of all training datas $H$ by using OMP algorithm
    for $k = 1$ to $K$ do
        - Calculate $E_k = H - \sum_{p \neq k} a_p x_p^T$
        - Define: $\omega_k = \{i | 1 \leq i \leq M, x_{k,i}^T \neq 0\}$
        - Let $E_k^R$ be the limited matrix of $E_k$ corresponding to $\omega_k$
        - Calculate SVD of $E_k^R$: $E_k^R = UDV$.
        - Update : $a_k = s_1$ and $x_k^R = d_1 v_1$ where $U = \{s_1, \ldots, s_L\}$, $V^T = \{v_1, \ldots, v_{|\omega_k|}\}$, $D = \text{diag}(d_1, d_2, \ldots, d_r)$, $r$ is the rank of the error matrix $E_k^R$
    end for
end while
$t = t + 1$;
OUTPUT: The result $A_{(t)}$
```

Table 1

Average AUC-PR values for the rotation and scale GREC dataset.

	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.2
(12, 3)	0.5331(-)	0.5908(-)	0.5950(-)	0.6073(=)	0.6105(=)	0.6064(=)	0.6146(=)	0.6119(=)	0.5995(-)	0.6028(-)	0.6015(-)
(12, 4)	0.3451(-)	0.4910(-)	0.5564(-)	0.5964(-)	0.6051(=)	0.6018(-)	0.6134(=)	0.6135(=)	0.6088(=)	**0.6199**	0.1061(-)
(12, 5)	0.2898(-)	0.4174(-)	0.5577(-)	0.5740(-)	0.5965(-)	0.6074(=)	0.6084(=)	0.6102(=)	0.6043(=)	0.6070(=)	0.6136(=)
(16, 3)	0.4205(-)	0.5379(-)	0.5803(-)	0.5935(-)	0.5981(-)	0.6008(-)	0.6041(-)	0.6012(=)	0.6030(-)	0.5949(-)	0.6101(=)
(16, 4)	0.3082(-)	0.4334(-)	0.5424(-)	0.5816(-)	0.5991(-)	0.6016(=)	0.6116(=)	0.6134(=)	0.6058(-)	0.6026(-)	0.6186(=)
(16, 5)	0.2108(-)	0.3474(-)	0.5040(-)	0.5678(-)	0.5838(-)	0.5960(-)	0.5932(-)	0.6075(=)	0.5838(-)	0.6101(=)	0.6163(=)

Table 2

Average AUC-PR values for the CVC dataset.

	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.2
(12, 3)	0.0766(-)	0.2062(-)	0.2360(-)	0.2710(-)	0.2920(-)	0.2561(-)	0.2498(-)	0.2860(-)	0.3100(-)	0.3026(-)	0.2958(-)
(12, 4)	0.0848(-)	0.1439(-)	0.2124(-)	0.2593(-)	0.2928(-)	0.3091(-)	0.3232(-)	0.3236(-)	0.3364(-)	0.3211(-)	0.3156(-)
(12, 5)	0.0831(-)	0.1301(-)	0.2211(-)	0.2887(-)	0.3095(-)	0.3106(-)	0.3471(-)	**0.3604**	0.3470(-)	0.3348(-)	0.3378(-)
(16, 3)	0.0719(-)	0.1791(-)	0.2266(-)	0.2584(-)	0.2316(-)	0.2078(-)	0.2222(-)	0.2847(-)	0.2887(-)	0.2873(-)	0.3034(-)
(16, 4)	0.0726(-)	0.1257(-)	0.1796(-)	0.2586(-)	0.2849(-)	0.3139(-)	0.3171(-)	0.3220(-)	0.3144 (-)	0.2873(-)	0.3034(-)
(16, 5)	0.0515(-)	0.0992(-)	0.1875(-)	0.2529(-)	0.3018(-)	0.3195(-)	0.3310(-)	0.3432(-)	0.3371(-)	0.3318(-)	0.3154(-)

Table 3

Average AUC-PR values for the Fresh dataset.

	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1	0.2
(12, 3)	0.3603(-)	**0.4018**	0.3643(-)	0.3692(-)	0.3656(-)	0.3615(-)	0.3622(-)	0.3649(-)	0.3580(-)	0.3653(-)	0.3576(-)
(12, 4)	0.3329(-)	0.3280(-)	0.3080(-)	0.3100(-)	0.300(-)	0.3317(-)	0.3290(-)	0.3345(-)	0.3417(-)	0.3039(-)	0.2983(-)
(12, 5)	0.2532(-)	0.2560(-)	0.2683(-)	0.2759(-)	0.2733(-)	0.2632(=)	0.2708(-)	0.2826(-)	0.2660(-)	0.2795(-)	0.2679(-)
(16, 3)	0.3211(-)	0.3033(-)	0.2787(-)	0.2848(-)	0.2750(-)	0.2802(-)	0.2808(-)	0.2802(-)	0.2800(-)	0.2814(-)	0.2503(-)
(16, 4)	0.2899(-)	0.2835(-)	0.2821(-)	0.2832(-)	0.2833(-)	0.2884(-)	0.2910(-)	0.3128(-)	0.2862(-)	0.2795(-)	0.2452(-)
(16, 5)	0.2560(-)	0.2620(-)	0.2469(-)	0.2376(-)	0.2359(-)	0.2359(-)	0.2441(-)	0.2465(-)	0.2400(-)	0.2448(-)	0.2243(-)

Table 4

Best values for the datasets.

Dataset	L	ϵ	Av. time (s)/query	Av. AUC-PR
GREC2003	48 (12 ×4)	0.1	1.380	**0.6199** (+)
CVC	60 (12 × 5)	0.08	0.831	**0.3604** (+)
FRESH	36 (12 × 3)	0.02	20.824	**0.4018** (+)

Table 5

Retrieval effectiveness with AUC-PR values in different datasets

	G_1 Zernike	G_2 Zernike	R-Signature	GFD	SC	SCIP	SIFT	SIFT+sparse	Our Approach (SCIP+sparse)
Rotation and Scaling	0.057	0.075	0.041	0.202	0.088	0.548	0.153	0.213	**0.620**
Distoration	0.661	0.504	0.519	0.638	0.699	0.761	0.447	0.497	**0.773**
Deform and Degrade	0.499	0.273	0.460	**0.530**	0.220	0.292	0.203	0.234	0.457
CVC	0.064	0.015	0.053	0.025	0.002	0.220	0.021	0.029	**0.360**
FRESH	0.266	0.222	0.343	0.314	0.301	0.286	0.355	**0.443**	0.402

Table 6

Average of the AUC-PR values considering several percentages of training set size.

Dataset	Descriptor + sparse	25%	50%	75%	100%
GREC2003	SCIP	0.610 (=)	0.592 (=)	0.610 (=)	0.620
	SIFT	0.211 (=)	0.203 (=)	0.209 (=)	0.213
CVC	SCIP	0.295(-)	0.306(-)	0.340(-)	0.360
	SIFT	0.024 (-)	0.027 (=)	0.028 (=)	0.029
FRESH	SCIP	0.386 (=)	0.387(=)	0.397(=)	0.402
	SIFT	0.429(-)	0.440(=)	0.436(=)	0.443

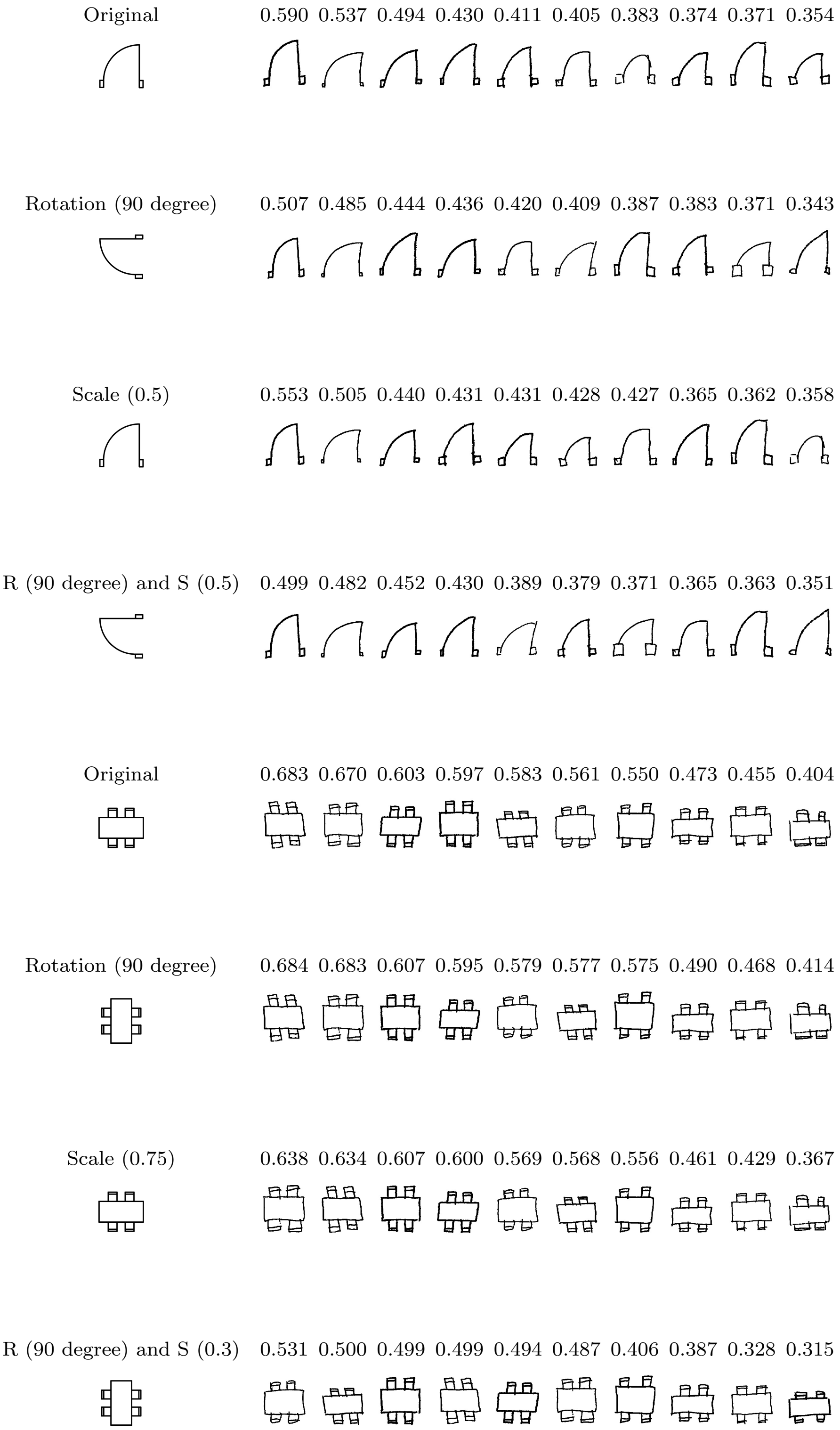
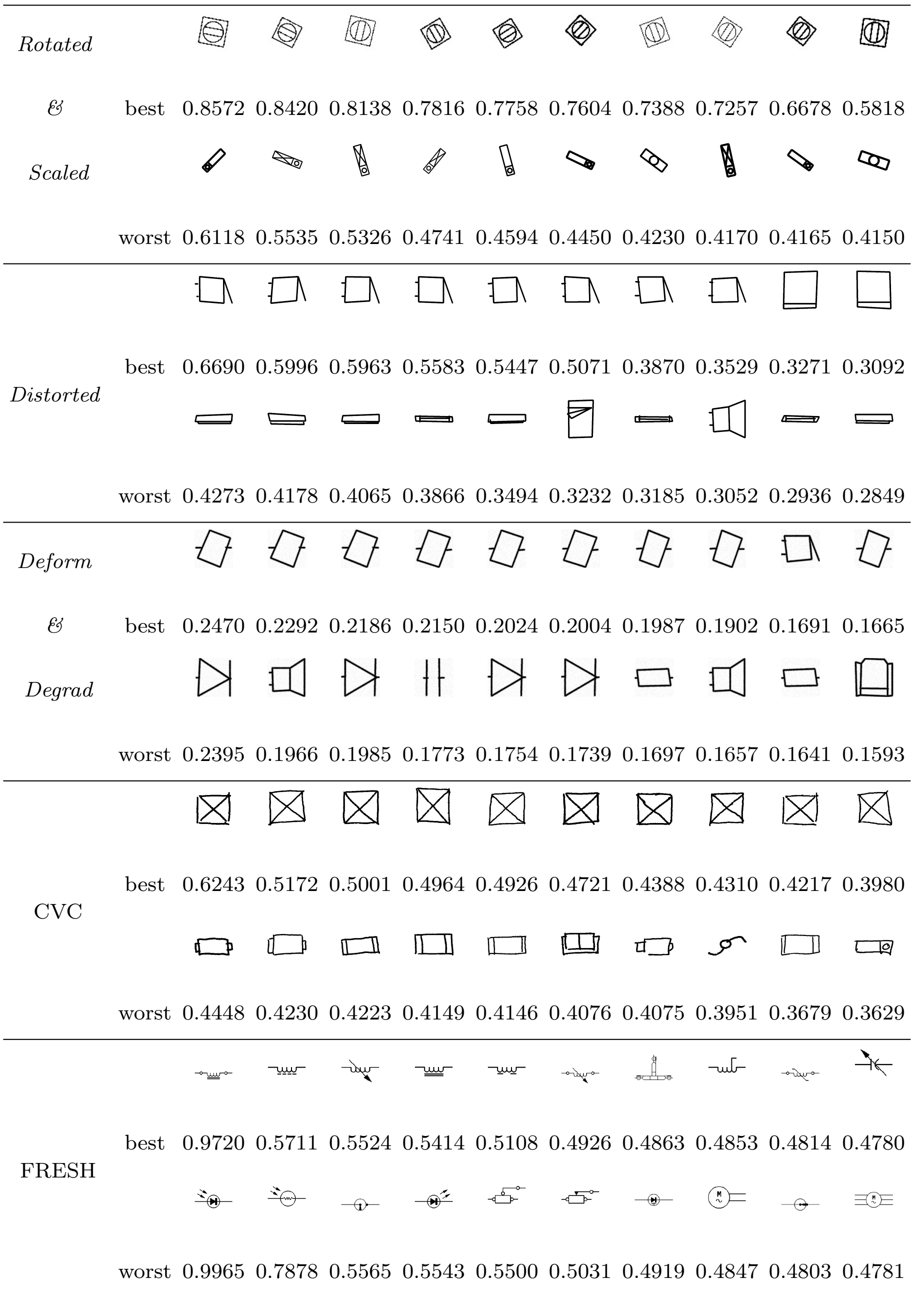


Fig. 3. Retrieval symbols (on CVC dataset) when we rotate, scale the query symbol (first column)



Rotated & Scaled	best	0.8572	0.8420	0.8138	0.7816	0.7758	0.7604	0.7388	0.7257	0.6678	0.5818
	worst	0.6118	0.5535	0.5326	0.4741	0.4594	0.4450	0.4230	0.4170	0.4165	0.4150
Distorted	best	0.6690	0.5996	0.5963	0.5583	0.5447	0.5071	0.3870	0.3529	0.3271	0.3092
	worst	0.4273	0.4178	0.4065	0.3866	0.3494	0.3232	0.3185	0.3052	0.2936	0.2849
Deform & Degrad	best	0.2470	0.2292	0.2186	0.2150	0.2024	0.2004	0.1987	0.1902	0.1691	0.1665
	worst	0.2395	0.1966	0.1985	0.1773	0.1754	0.1739	0.1697	0.1657	0.1641	0.1593
CVC	best	0.6243	0.5172	0.5001	0.4964	0.4926	0.4721	0.4388	0.4310	0.4217	0.3980
	worst	0.4448	0.4230	0.4223	0.4149	0.4146	0.4076	0.4075	0.3951	0.3679	0.3629
FRESH	best	0.9720	0.5711	0.5524	0.5414	0.5108	0.4926	0.4863	0.4853	0.4814	0.4780
	worst	0.9965	0.7878	0.5565	0.5543	0.5500	0.5031	0.4919	0.4847	0.4803	0.4781

Fig. 4. Examples of querying symbols achieving the best and worst retrieval results in terms of AUC-PR values.