Text/graphic separation using a sparse representation with multi-learned dictionaries

Thanh Ha Do, Salvatore Tabbone, Oriol Ramos Terrades

Text/graphic separation using a sparse representation with multi-learned dictionaries

Thanh-Ha Do*, Salvatore Tabbone*, Oriol Ramos-Terrades**
*Université de Lorraine-LORIA UMR 7503, Campus scientifique - BP 239, France
**Universitat Autonoma de Barcelona - Computer Vision Centre, Espanya
{ha-thanh.do, tabbone}@loria.fr; oriolrt@cvc.uab.cat

Abstract

In this paper, we propose a new approach to extract text regions from graphical documents. In our method, we first empirically construct two sequences of learned dictionaries for the text and graphical parts respectively. Then, we compute the sparse representations of all different sizes and non-overlapped document patches in these learned dictionaries. Based on these representations, each patch can be classified into the text or graphic category by comparing its reconstruction errors. Same-sized patches in one category are then merged together to define the corresponding text or graphic layers which are combined to create a final text/graphic layer. Finally, in a post-processing step, text regions are further filtered out by using some learned thresholds.

1. Introduction

Extracting text regions and other graphical objects is an important step of any automated document analysis system. The information about the type of extracting data can help to improve the accuracy rate as well as to speed up the recognition process. In addition, if the text is well segmented an OCR could be applied to define the semantic meaning of the extracted text regions. In this purpose, a great number of studies have been proposed to tackle the problem of text regions recognition in technical documents [1, 3, 4, 5, 6, 8]. In [6] the directional morphological operator has been proposed to deal with simple maps. However, the method is not robust for characters with different sizes included into the same document. The methods described in [1, 5] are dedicated to extract texts in engineering drawings, but they have some limitations linked to the kind of engineering drawings. The performance of the recognition rate in [5] depends strongly on nine preset thresholds. Moreover, the recursive merging algorithm *charbox growing* in [1] cannot work well in the case the text characters touch either themselves or the graphics. The algorithm reported in [3] is one of the well-known approaches based on the analysis of the connected component and the Hough transform to group together components into logical string of characters. This algorithm is simple and scalable, but its application to cluttered documents is difficult especially when texts touch the graphic. Tombre *et al* [8] made this method more suitable by choosing right thresholds and proposing a new post-processing step. The recent work in [4] proposes to segment technical documents into two morphological components using the Morphological Component Analysis (MCA) framework with two pre-constructed dictionaries. However, as the results are dependent on the choice of two pre-constructed dictionaries, this method is not adapted to various kind of documents.

In summary, the existing methods are not efficient when dealing with documents containing dense information. Therefore, in this paper, we propose an alternative approach that overcomes the limitations of existing methods by using a multi-learned dictionaries combined with a sparse representation. In fact, learned dictionaries were used successfully in local/global MCA with the purpose of separating textures and cartoons [2], as well as in the text detection from scenic images [7]. However, to the best of our knowledge, the learned dictionary, especially multi learned dictionaries have never been used for separating the text regions from graphical part.

The main idea of the proposed method is based on the assumption that the representation of text candidate patches in the learned dictionaries for texts are sparse but not sparse in the learned dictionaries for graphics. To make use of this assumption, we first empirically construct two sequences of dictionaries corresponding to two types of data (graphic or text) using the K-SVD

method [2]. Then, we use these learned dictionaries combined with Orthogonal Matching Pursuit (OMP) algorithm [2] to find the sparse representations of all different sized non-overlapped patches. Next, each patch can be classified into text or graphic categories by comparing its reconstruction errors. All same-sized patches in one category are merged to make a corresponding text or graphic layers. Finally, these text/graphic layers are combined by using logical operators. In a post processing step, text regions are further filtered out by using some learned thresholds.

The rest of this paper is organized as follows. In Section 2, the sparse representation is recalled. A discussion about the learning method K-SVD as well as how to use it to create multi-learned dictionaries for text extraction is given in Section 3. Details of the proposed text detection method are presented in Section 4. The experimental results show that our method provides good results and outperforms other methods (Section 5). Finally, we give our conclusions in the Section 6.

2. Sparse Representation

The problem solved by the sparse representation is to find the most compact representation of an image in terms of a linear combination of atoms in an overcomplete dictionary. In other words, this problem can be formulated as follows. Given a matrix $A = \{a_1, a_2, ..., a_m\} \in \mathbb{R}^{n \times m}$ and a signal $y \in \mathbb{R}^n$, we consider the underdetermined linear system of equations $y = Ax$ with $m \gg n$. If A is a full$-$rank matrix, there will be infinitely many different sets of values for the x_i's that satisfy all equations simultaneously. To find one well-defined solution, one solution can explain the signal well comparing with others, a function $f(x)$ is added to assess the desirability of a would-be solution x, with smaller values being preferred. In the case $f(x)$ is the l_0 pseudo-norm $\|x\|_0$ (number nonzero elements in vector x), the well-defined solution is the solution of the equation (P_0).

$$(P_0): \ \bar{x} = \arg\min_x \|x\|_0 \text{ s.t } Ax = y \qquad (1)$$

In general, solving equation (1) is often difficult (NP-hard problem), and *Donoho et al* [2] approached the P_0 problem by substituting it by a convex relaxation instead with some appropriate conditions on A and x, such as $\|x\|_0 = k_0 \leq spark(A)/2$ [1], to guarante that the solution of the following equation P_1 is unique and also the unique solution of P_0.

$$(P_1): \ \bar{x} = \arg\min_x \|W^{-1}x\|_1 \text{ s.t } Ax = y \qquad (2)$$

The matrix W is a diagonal positive-definite matrix, defined by $w(i,i) = 1/\|a_i\|_2$. The solution of the equation (1) or (2) contains the representation coefficients of the signal y and is called the sparse representation of y.

3. Learned Dictionaries for Text Detection

3.1 K-SVD Algorithm

In K-SVD algorithm, a family l signals $\{y_j\}_{j=1}^l$ is considered as the training database. Our goal is to find a dictionary A in which each signal $y_j \in \mathbb{R}^n$ has an optimal sparse approximation:

$$\min_{x_j, A} \sum_{j=1}^{l} \|y_j - Ax_j\|_2^2 \text{ s.t } \|x_j\|_0 \leq T_0, j = 1, ..., l \quad (3)$$

Such dictionary can be obtained by the learning process that iteratively adjusts A via two main steps. In the first step, all sparse representations $X = \{x_j\}_{j=1}^l$ of $Y = \{y_j\}_{j=1}^l$ are found under the condition of A is fixed. In the second step, an updating rule, that makes a modification sequentially on each column a_{j_0} of A, is applied to optimize the residual error in this equation:

$$\|Y - AX\|_F^2 = \|E_{j_0} - a_{j_0} x_{j_0}^T\|_F^2 \qquad (4)$$

where X, $\{a_1, ..., a_{j_0-1}, a_{j_0+1}, ..., a_l\}$ are fixed, and $E_{j_0} = Y - \sum_{j \neq j_0} a_j x_j^T$.

In this description, $x_{j_0}^T$ is the j_0-th row of X and the notation $\|.\|_F$ stands for the Frobenius norm. Since $\{a_1, ..., a_{j_0-1}, a_{j_0+1}, ..., a_l\}$ are fixed, then E_{j_0} is fixed. The minimization error $\|Y - AX\|_F^2$ depends only on the optimal values of a_{j_0} and $x_{j_0}^T$. These optimal solutions can be obtained with the *Singular Value Decomposition* (SVD) of the matrix E_{j_0}. However, in this case, there is no way to guarantee that the number of non-zeros in the $x_{j_0}^T$ is lower, or in other words, the condition about the sparsity of X can be broken. This problem can be overcome by calculating only the SVD of the sub-matrix of E_{j_0} that includes the columns where the entries in the row $x_{j_0}^T$ are non-zeros. More details about K-SVD algorithm can be found in [2].

3.2 Learned Dictionaries for Text Extraction

In this paper, two sequences of dictionaries are used for text and graphic separation, each sequence has K different dictionaries. To create the sequences of

[1] *spark*(A): is the smallest number of columns that are linearly-dependent.

the text and graphics dictionaries, named $\{A_k\}_{k=1}^K$, $\{B_k\}_{k=1}^K$, respectively, first of all we need to create the corresponding sequences of training databases $\{Y_k\}_{k=1}^K$ and $\{Z_k\}_{k=1}^K$. Columns of Y_k, Z_k are composed of non-overlapped patches with a size $\sqrt{s_k} \times \sqrt{s_k}$ extracted from training examples of texts and graphics. The text training examples are composed by 26 lowercase and uppercase English letters and 10 Arabic numerals of various fonts, sizes and types and their 90 degrees clockwise rotation. The graphics training examples are collected from the graphic images including only the graphic component and their different resolutions.

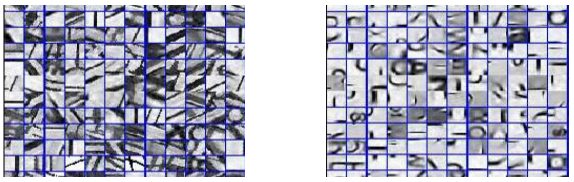

Figure 1. A zoom of the trained dictionaries with $\sqrt{s_k} = 16$: Graphic (left) and Text (right)

.

After getting two sequences of the training databases, we apply the K-SVD algorithm to construct $\{A_k\}_{k=1}^K, \{B_k\}_{k=1}^K$. Each dictionary A_k (or B_k) is a matrix with s_k rows and $4 \times s_k$ columns (see figure 1).

4. Detection of Text Regions via Sparse Representation

Given a graphical image $y \in \mathbb{R}^{n \times m}$, we first decompose it into K sets of non-overlapped patches by using K sliding windows $\{w_k\}_{k=1}^K$, in which w_k has the size $\sqrt{s_k} \times \sqrt{s_k}$. Afterwards, for each set of patches $\{p_i^k\}_i$, we use two learned dictionaries A_k, B_k combined with OMP [2] to find all sparse representations of all patches in this set, named $\{\bar{q}_{a,i}^k\}_i$, here a stands for A_k and B_k:

$$\bar{q}_{a,i}^k = \arg\min_{q_{a,i}^k} \|p_i^k - a q_{a,i}^k\|_2 \text{ s.t } \|q_{a,i}^k\|_0 \leq T_k \quad (5)$$

Then, each patch p_i^k can be classified into text or graphic by comparing its reconstruction errors in A_k, B_k using equation (6).

$$\epsilon_{a,i}^k = \|p_i^k - a\bar{q}_{a,i}^k\|_2 \quad (6)$$

If the reconstruction error of p_i^k in text learned dictionary A_k is smaller than its reconstruction error in the graphic learned one, B_k, it means that the representative of p_i^k in A_k is sparse and not sparse (or at least not enough sparse) in B_k and, p_i^k is considered as a text patch. Otherwise, it is classified as a graphic patch.

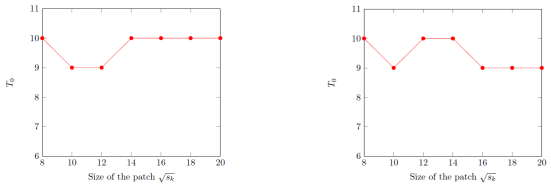

Figure 2. The optimal value of T_0 following the size of the patch for the text dictionaries (left) and graphic dictionaries (right) in term of average representation error.

Next, all text/graphics patches are added to the text/graphics layer y_T^k, y_G^k, respectively. The K text and graphic layers ($K = 2$ in our case) are combined into the final text/graphic layer by using the logical operations $y_T = \wedge y_T^k$ and $y_G = \vee y_G^k$.

The post processing phase is necessary to further filter out some text candidates. To delete some remaining graphic components, we learn thresholds defined on the behavior of the sparsity of noise components related to real true texts components. More details are given in the next section.

5. Experimental Results

In the K-SVD algorithm, we have to take care on the value T_0 and the size of the patch which have an impact on the learned dictionaries. We can say that if the size of the patch is too small, the information in the graphic and text patch is not so much different, so, text candidate patches can be considered as graphic patches and reciprocally. If the size is too large, each patch can contain the text and graphic components together, and there is not enough sparsity, either in text dictionary or in graphic dictionaries. Moreover, if we analyze the bahavior of T_0 we can remark that the optimal value of this threshold is different following the size of the chosen patch (see figure 2).

In this perspective, we propose the use of multi-learned dictionaries as describe in Section 4. We generate a set of dictionaries for $\sqrt{s_k}$ from 8 to 22 and using a sequential forward selection algorithm. We experimentally find that the best trade-off is the combination of two dictionaries with $\sqrt{s_k} = 8$ and 16.

In the final text layer, there are still some graphical components that are considered as noise components so far. This kind of noise will be deleted by verifying the behavior of its sparsity and compared with the one of the true texts (characters) in the text/graphical dictio-

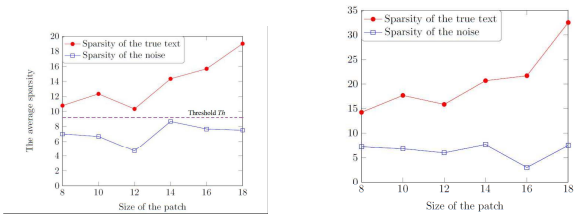

Figure 3. Behaviour of the sparsity of the texts and the noise components in the text dictionaries (left) and graphic dictionaries (right).

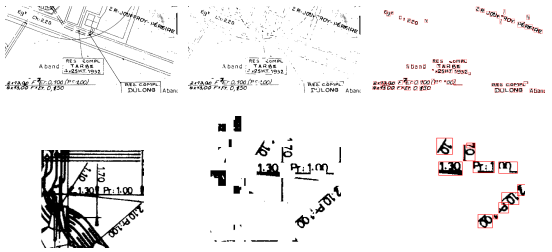

Figure 4. Examples of documents used in the evaluation of table 1: Origial images (left), Text layers (midle), Text extraction (right).

naries. Figure (3) shows the average sparsity of the non-overlapped patches of the noise components and the text components in the text and graphic dictionaries. The figure clearly shows that, in the text dictionaries, the sparse representation of the noise component has less non-zero elements than the text. Moreover, the sparse representation of the noise component in the text dictionaries is sparser than in the graphic dictionaries. This explains why some noise components are misclassified may as text candidates. In this perspective, we consider that a patch is a text if its sparsity is above a threshold *Th*. This threshold is larger than the average sparity of the remained noise components, see figure 3 (left).

We compare our method with the one proposed by Thai *et al* [4] and Tombre *et al* [8], using the same quantitative measures where Nb. ch is the number of characters in each image counted by the same protocol as in [8]. From Table (1) we can remark that our method provides the best results for each document and is better than each dictionary used separately.

Table 1. Performance evaluation: (see Fig. 4), with T_0 set in Figure 2 and $T_k = 16; 32$ for $\sqrt{s_k} = 8; 16$.

Img	Nb. ch.	[4]	[8]	Our method	Dic. 8×8	Dic. 16×16
Doc 1	63	53	58	61	24	50
Doc 2	92	70	71	85	38	78
Doc 3	93	77	81	86	32	83
Doc 4	121	104	104	114	53	111
Doc 5	31	22	7	23	6	19

6. Conclusion

In this paper, we present an alternative approach for text detection from technical documents based on sparse representation. In our method, we combine multi-trained dictionaries and sparse representation. The experimental results show that this combination could be a good choice for the segmentation problem with complex graphical documents. We propose an original way to set the sparse thresholds automatically. Additionally, we overcome the restrictions of the existing methods to some kind of document only (same orientation of the text and same font size).

References

[1] D. Dori and L. Wenyin. Vector-based segmentation of text connected to graphics in engineering drawings. *Advances in structural and syntactical pattern recognition*, 1121:322–331, 1996.

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

[3] L. A. Fletcher and R. Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):910–918, 1988.

[4] T.-V. Hoang and S. Tabbone. Text extraction from graphical document images using sparse representation. *International workshop on Document Analysis Systems*, pages 143–150, 2010.

[5] Z. Lu. Detection of text regions from digital engineering drawings. *Pattern Analysis and Machine Intelligence*, 20(4):431–439, 1998.

[6] H. Luo and R. Kasturi. Improved directional morphological operations for separation of characters from maps/graphics. *Graphics recognition algorithms and systems*, 1389:35–47, 1998.

[7] W. Pan, T. D. Bui, and C. Y. Suen. Text detection from scene images using sparse representation. *Proceedings of the 19th International Conference on Pattern Recognition*, pages 1–5, 2008.

[8] K. Tombre, S. Tabbone, L. Pélissier, B. Lamiroy, and P. Dosch. Text/ graphics separation revisited. *DAS'02 proceedings of the 5th International Workshop on Document Analysis Systems*, 2432(1):200–211, 2002.