Text extraction from graphical document images using sparse representation

Thai V. Hoang, Salvatore Tabbone

Text Extraction From Graphical Document Images Using Sparse Representation

Thai V. Hoang[*,**], Salvatore Tabbone[*]
[*] MICA, UMI 2954, Hanoi University of Technology, Hanoi, Vietnam
[**] LORIA, UMR 7503, Université Nancy 2, 54506 Vandoeuvre-lès-Nancy, France
vanthai.hoang@loria.fr tabbone@loria.fr

ABSTRACT
A novel text extraction method from graphical document images is presented in this paper. Graphical document images containing text and graphics components are considered as two-dimensional signals by which text and graphics have different morphological characteristics. The proposed algorithm relies upon a sparse representation framework with two appropriately chosen discriminative overcomplete dictionaries, each one gives sparse representation over one type of signal and non-sparse representation over the other. Separation of text and graphics components is obtained by promoting sparse representation of input images in these two dictionaries. Some heuristic rules are used for grouping text components into text strings in post-processing steps. The proposed method overcomes the problem of touching between text and graphics. Preliminary experiments show some promising results on different types of document.

Categories and Subject Descriptors
I.4.6 [**Image Processing and Computer Vision**]: Segmentation; I.5.4 [**Pattern Recognition**]: Applications—*text processing*; I.7.1 [**Document and Text Processing**]: Document Capture—*document analysis, graphics recognition and interpretation*

General Terms
Design, Documentation, Experimentation

Keywords
Text/graphics separation, sparse representation, morphological component analysis, curvelet transform, redundant wavelet transform, text component grouping

1. INTRODUCTION
Text extraction from graphical document images is a major problem in document image analysis in which one document input image containing both text and graphics is processed to produce two output images, one containing text and the other containing graphics. Extracting text is an important task since text has semantic meaning which could be obtained by a character recognition system. Extraction of semantic meaning from the extracted text can be done easily with the help of an OCR engine. A reliable extraction method is required to make it usable in automatic document processing systems. At present, applications of such text extraction algorithm are automatic processing of texture documents and architectural/engineering drawings, automatic reading of postal addresses and flexible forms, *etc.*

The remainder of this paper is organized as follows. Section 2 reviews some related works and briefly highlights the contribution of the proposed method. Some background on sparse representation of signals is given in Section 3. Section 4 presents the decomposing algorithm and dictionary selection. Post-processing steps on the text image are discussed in Section 5. Experimental results are given in Section 6, and finally conclusions are drawn in Section 7.

2. THE PROBLEM OF TEXT EXTRACTION FROM GRAPHICAL DOCUMENT IMAGES
A document image usually contains text and graphics components. The types of graphics component vary according to each specific application domain but generally they include lines, curves, polygons, circles. Text components consist of characters and digits which form words and phrases used to annotate the graphics. Extraction of text components is a challenging problem because:

- Graphical components like lines can be of any length, thickness, and orientation. Circles, polygons can be filled or unfilled. Text components can vary in font styles and sizes.
- There may exist touching among text components and touching, crossing between text and graphics components. Text strings are usually intermingled with graphics and are of any orientation.
- Excluding the pre-processing steps to enhance the image quality, text extraction is mostly the first step in the chain of document analysis with limited knowledge about the presence of high-level objects in the images.

2.1 Prior work
Many methods have been proposed to tackle the problem of text extraction from graphics-containing documents which

DAS '10, June 9-11, 2010, Boston, MA, USA

can be roughly divided into three main families: morphological analysis, connected component analysis, and multi-resolution analysis.

Run-Length Smoothing Algorithm (RLSA) proposed by Wahl *et al.* [33] is one of the first and best known methods based on morphological filtering to detect long vertical and horizontal text strings. It essentially consists of morphological closing operations with horizontal and vertical structuring elements of specified length. Although RLSA and its improvement [19] are very efficient for textual documents, its use in graphics-rich documents [18, 20] is limited as text could be wrongly labeled as graphics.

A well-known approach based on connected component analysis was proposed by Fletcher and Kasturi [12] using some heuristic rules on area, dimension ratio and collinearity of connected components to separate text from graphics. Simplicity and scalability are the strength of this approach which make it widely used. Its weakness is the inability to directly separate text which touches graphics. Tombre *et al.* [32] tried to overcome this and got some improvements for graphics-rich document images by incorporating some more heuristic rules.

Multi-resolution approach was first proposed by Deforges and Barba [7] for mail pieces and then adapted to map by Tan and Ng [31]. It relies on the assumption that at a certain coarse level of the image pyramid, a text line looks like a long component and at the next finer level it looks like a regular sequence of transitions. However, when text and graphics components lie closely or touch, this approach induces wrong detection results.

Some methods have been proposed dealing with the case of touching between text and graphics components. They usually separate text from graphics by detecting touching lines. Gloger [14] uses Hough transform to detect vertical and horizontal lines to recognize form structure. Lu [20] detects slant lines in engineering drawings by first stretching the document to certain angles and then tracing black pixels horizontally and vertically. Luo and Kasturi [21] use directional morphological filtering to locate linear shapes in simple maps. Cao and Tan [4] work on a skeletonized version of a map and consider short and long skeleton segments as skeletons of text and graphics components respectively. Recently, Su *et al.* [29] rely on a vectorization method [27] to remove touching lines in engineering drawings.

Each of the above methods dealing with touching is initially designed for a specific application so it is not robust and inapplicable to graphical images from other applications. For example, with a graphics-rich and complex engineering document image as showed in Fig. 1(a), none of the above methods provide reliable results. Thus a novel method needs to be developed for these kinds of complex graphical document.

2.2 The proposed method

The method proposed in this paper extracts texts components in a totally different way from the above. A document image y containing text and graphics components is considered as a two-dimensional signal which is the mixture of two separate two-dimensional signals (images) of the same size: y_t containing text components and y_g containing graphics components. The problem of text extraction is now seen as the problem of recovery of y_t and y_g from y. This is actually the blind source separation problem in multi-dimensional signal processing [17].

To solve this, we employ the Morphological Component Analysis (MCA) method proposed by Starck *et al.* [28]. MCA allows the separation of features contained in an image when these features present different morphological aspects. This is facilitated by promoting sparse representation of these features in two appropriately chosen dictionaries, each leads to sparse representation over one feature and non-sparse representation over the other.

Having done in this way, some post-processing steps could be needed to extract text strings from y_t. This is done with the help of some heuristic rules based on the discriminative characteristics of text components. The proposed method is thus robust to touching between text and graphics. Furthermore, text components can be placed anywhere with any orientation in the documents. They can be in any form, have any font style and size.

3. SPARSE REPRESENTATION OF SIGNALS OVER AN OVERCOMPLETE DICTIONARY

The idea of sparse representation has its root in mammalian vision system. The receptive fields of simple cells in mammalian primary visual cortex can be characterized as being spatially localized, oriented, and bandpass (selective to structure at different scales), comparable to the basis functions of wavelet transforms, and having a strategy for producing a sparse distribution of output activity in response to natural images [22]. Olshausen and Field [23] validated this theory by considering the problem of efficient coding of natural images. They showed that when the code dictionary is overcomplete (the number of code elements is greater than the dimensionality of the input) and non-orthogonal, a coding strategy maximizing sparseness (a small number of code elements are non-zero) will select only codes necessary for representing a given input. According to Barlow's principle of redundancy reduction [2], the resulting sparse code thus provides an efficient representation for later processing.

The above idea can be formulated mathematically. Given an input image y of size $w \times h$ which is casted as a vector $b \in \mathbb{R}^n (n = wh)$ by stacking its columns, an overcomplete dictionary $A \in \mathbb{R}^{n \times K}$ with $n \ll K$ allowing sparse representation of b, let x be the representation of b in A satisfying $b = Ax$ then finding the sparse representation $\hat{x}$ of b in A is equivalent to solving the following ℓ_0-optimization problem:

$$\min_x \|x\|_0 \quad \text{subject to} \quad Ax = b \tag{1}$$

where $\|.\|_0$ is the ℓ_0 norm, counting the nonzero entries of a vector. The overcomplete dictionary A that leads to sparse representation of an input signal b can either be designed by adapting its content to fit a given set of signal examples [1] or chosen as a specified set of signals by means of undecimated wavelet transform, Fourier transform, short-time Fourier transform, Gabor transform, curvelet transform, steerable wavelet transform, *etc.*

Finding the sparsest solution of an under-determined system of linear equations like (1) is NP-hard [6]. However, recent results [8] show that if the solution of Eq. (1) is sufficiently sparse, it is equal to the solution of the following ℓ_1-optimization problem:

$$\min_x \|x\|_1 \quad \text{subject to} \quad Ax = b. \tag{2}$$

If the condition $b = Ax$ is relaxed by $b = Ax + z$, where $z \in \mathbb{R}^n$ is a noise term with $\|z\|_2 < \varepsilon$, to account for the possible inclusion of small dense noise in the input image y or allow small error in the representation, Eq. (2) will be modified to be:

$$\min_x \|x\|_1 \quad \text{subject to} \quad \|Ax - b\|_2 \leq \varepsilon. \tag{3}$$

This is a convex optimization problem, its solution can be found by minimizing the corresponding Lagrangian function:

$$\min_x \frac{1}{2}\|Ax - b\|_2 + \lambda\|x\|_1, \tag{4}$$

where the parameter λ is a Lagrange multiplier that balances the sparseness in $\hat{x}_1$ and the representation error, the value of λ depends on ε. This problem is known to be efficiently solved via second-order cone programming [5].

4. SEPARATION OF TEXT AND GRAPHICS COMPONENTS USING MORPHOLOGICAL COMPONENT ANALYSIS

MCA method is a further development of the framework represented in the previous section dealing with the problem of separating an image content into semantic parts. MCA has shown to be very useful for decomposing images into texture and piece-wise smooth (cartoon) parts or for inpainting applications [10, 11]. In document image analysis, MCA has been adopted by Pan *et al.* [24] to segment text from complex background. The application of MCA to a new application domain, text/graphics separation, is presented in this paper and is followed by some post-processing steps proposed uniquely for this kind of application.

4.1 Morphological Component Analysis

Let a signal $b \in \mathbb{R}^n$ be a linear combination of two parts, $b = b_1 + b_2$ where b_1 and b_2 represent two different types of signal. Assume that there exist two overcomplete dictionaries $A_1, A_2 \in \mathbb{R}^{n \times K}$ satisfying two conditions:

1. Solving (for $i = 1, 2$)
$$\min_{x_i} \|x_i\|_1 \quad \text{subject to} \quad A_i x_i = b_i, \tag{5}$$
leads to a sparse representation $\hat{x}_i$ of b_i in A_i.
2. Solving (for $i \neq j$)
$$\min_{x_i} \|x_i\|_1 \quad \text{subject to} \quad A_j x_i = b_i, \tag{6}$$
leads to a non-sparse representation $\hat{x}_i$ of b_i in A_j.

In this case, two dictionaries A_1, A_2 are said to be discriminative in the sense of sparse representation to different content types. MCA method thus proposes to solve the following optimization problem:

$$\min_{x_1, x_2} (\|x_1\|_0 + \|x_2\|_0) \quad \text{subject to} \quad A_1 x_1 + A_2 x_2 = b, \tag{7}$$

which can be converted to:

$$\min_{x_1, x_2} (\|x_1\|_1 + \|x_2\|_1 + \lambda\|b - A_1 x_1 - A_2 x_2\|_2). \tag{8}$$

Solving Eq. (8) gives $\hat{x}_1$ and $\hat{x}_2$, the sparse representations of b_1 and b_2 in A_1 and A_2 respectively. This also means that the original signal b has been separated into two parts $A_1\hat{x}_1$ and $A_2\hat{x}_2$ which are in turn the approximations of b_1 and b_2 respectively. For this problem structure, Block-Coordinate Relaxation (BCR) method by Sardy *et al.* [25] provides fast numerical computation that requires only the use of matrix-vector multiplications with the unitary transforms and their inverses. BCR was developed based on the shrinkage method of Donoho and Johnstone [9].

4.2 Dictionary selection

The success of MCA is guaranteed if the two conditions stated in Eqs. (5) and (6) are satisfied. Thus, selecting two appropriate dictionaries A_1 and A_2 is essential in applying MCA for signal separation. For numerical reasons, A_1 and A_2 should also have fast forward and inverse implementations. Our approach here is to choose these dictionaries from existing transforms based on experience. Curvelet transform [3] is used as the dictionary for graphics and the undecimated wavelet transform is used as the dictionary for text.

4.2.1 Undecimated wavelet transform

Undecimated wavelet transform (UWT) is the undecimated version of the orthogonal wavelet transform (OWT) obtained by skipping the decimation step. It is designed to overcome the lack of shift-invariance property in OWT. UWT, not like OWT, can be represented as a transformation matrix with more columns than rows. The redundancy factor (the ratio between the number of columns to the number of rows) is $3J + 1$, where J is the number of scales. UWT is expected to give sparse representation of isotropic features and non-sparse representation to highly anisotropic features. The "à trous" algorithm by Shensa [26] provides an efficient way to implement forward and inverse UWT.

4.2.2 Curvelet transform

Defining two smooth, non-negative, and real-valued functions $W(r)$ and $V(t)$ (the radial angular windows) which are supported on $[1/2, 2]$ and $[-1, 1]$ respectively satisfying the admissibility conditions with $r > 0$ and $t \in \mathbb{R}$:

$$\sum_{j=-\infty}^{\infty} W^2\left(2^j r\right) = 1, \quad \sum_{l=-\infty}^{\infty} V^2\left(t - l\right) = 1. \tag{9}$$

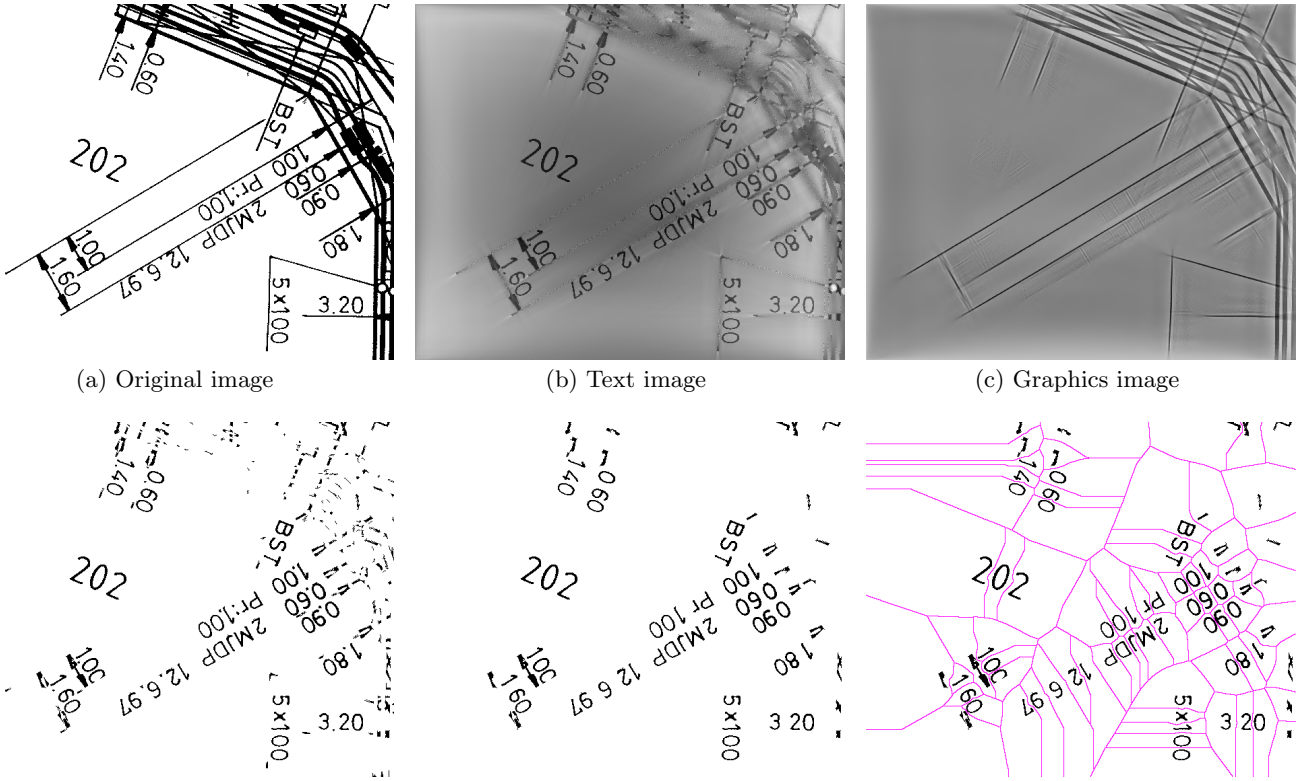
At each scale j, define the mother curvelet φ_j as:

$$\hat{\varphi}_j(r, \theta) = 2^{-3j/4} W\left(2^{-j} r\right) V\left(\frac{2^{\lfloor j/2 \rfloor}\theta}{2\pi}\right). \tag{10}$$

A curvelet at scale j, orientation θ_l, and position $\mathbf{x}_\mathbf{k}^{j,l} = R_{\theta_l}^{-1}\left(2^{-j}k_1, 2^{-j/2}k_2\right)$ is defined as:

$$\varphi_{j,l,\mathbf{k}}(\mathbf{x}) = \varphi_j\left(R_{\theta_l}\left(\mathbf{x} - \mathbf{x}_\mathbf{k}^{j,l}\right)\right), \tag{11}$$

where R_{θ_l} is the rotation operator by $\theta_l = 2\pi 2^{\lfloor j/2 \rfloor} l$ radians with $l \in \mathbb{Z}^+$ such that $0 \leq l < 2\pi$. The corresponding



(a) Original image (b) Text image (c) Graphics image

(d) Binarized text image (e) Binarized text image after removing small components (f) Area Voronoi diagram

Figure 1: Processing steps to extract text strings from an image

curvelet coefficients of $f \in L^2\left(\mathbb{R}^2\right)$ is defined as the inner product:

$$c_{j,l,\mathbf{k}} = \langle f, \varphi_{j,l,\mathbf{k}} \rangle = \int_{\mathbb{R}^2} f(\mathbf{x}) \overline{\varphi_{j,l,\mathbf{k}}(\mathbf{x})} d\mathbf{x}. \qquad (12)$$

Curvelets constructed in this way are multi-scale, multi-directional, and elongated. They define a tight frame in $L^2\left(\mathbb{R}^2\right)$, exhibit an oscillating behavior in the direction perpendicular to their orientation, and obey the parabolic scaling relation ($width = length^2$). A curvelet frame can be used as an overcomplete dictionary with a redundancy factor of $16J+1$, where J is the number of scales. It is expected to give sparse representation to anisotropic structures and smooth curves and edges of different lengths.

4.3 Text image extraction

Supposed that an input document image y can be decomposed into two images of the same size as y: y_t containing text components and y_g containing graphics components. Applying MCA on y with UWT and curvelet transform as the two overcomplete dictionaries will result in $\tilde{y}_t$ and $\tilde{y}_g$ which are approximations of y_t and y_g respectively. Assuming that y takes the graphical image in Fig. 1(a), $\tilde{y}_t$ and $\tilde{y}_g$ are given in Figs. 1(b) and 1(c) respectively.

The obtained results in Figs. 1(b) and 1(c) show that text and graphics components are not totally separated. There are two reasons for this:

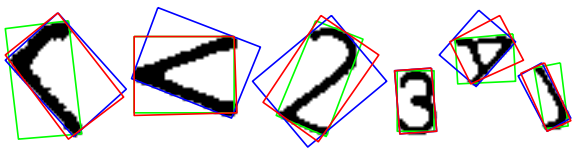
- There exists an overlap between the two dictionaries, both consider the low-frequency content as theirs and both can represent it efficiently.
- Some graphics (like arrowheads, short curve segments) have morphological characteristics that are similar to those of text components. These graphics may appear in the text image.

To minimize the effect of these ambiguities (i.e. the overlapping between the two dictionaries and the similarity between features), post-processing steps presented in the next section are proposed to combine the extracted text components into text strings.

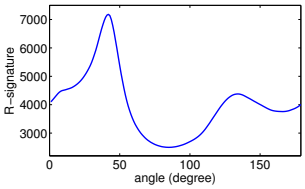
5. GROUPING TEXT COMPONENTS INTO TEXT STRINGS

The text image in Fig. 1(b) is converted to binary in Fig. 1(d) by adaptive thresholding [15]. Fig. 1(e) is obtained by removing small connected components from Fig. 1(d). It is acknowledged that small text components like '·', ':' are also removed, however, as these components lie inside the enclosing rectangles of text strings, they can be retrieved later.

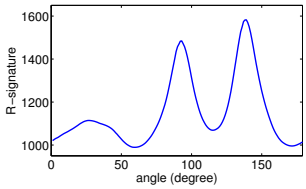
Connected components remaining in Fig. 1(e) are not only text components, they may be parts of graphics. Thus, an algorithm to group text components into text strings is required for a successful extraction of text components. Al-



(a) Enclosing rectangles of some characters



(b) R-signature of '`r`'



(c) R-signature of '`A`'

Figure 2: Determination of orientation

gorithms based on Hough transform on the centroids on connected components [12, 32] can be used in this case. However, to be robust, the grouping algorithm should be dependent on the style of text that exists in graphical document images. We propose here a new efficient method to group text components in straight fonts with a belief that a large part of text in graphical document images is typeset in straight font style. Heuristic criteria used are: *neighborhood*, *inter-distance*, *orientation*, and *overlapping*.

Neighborhood. Text components belonging to a text string need to be neighbors continuously. The neighborhood is determined by means of an area Voronoi diagram [16]. Fig. 1(f) shows the area Voronoi diagram of the binary image in Fig. 1(d). Each connected component is represented by one Voronoi region containing points that are closer to that connected component than to any other. Two text components are said to be neighbors if their representing Voronoi regions are adjacent.

Inter-distance. The inter-distance $d(g_i, g_j)$ between two text components g_i, g_j in one text string defined as:

$$d(g_i, g_j) = \min_{p \in g_i, q \in g_j} d(p, q) \tag{13}$$

should have restriction depending on their height $h(g_i)$, $h(g_j)$ as $d(g_i, g_j) < T_d \max\{h(g_i), h(g_j)\}$. The value of T_d is determined by experience and is equal to 1.2.

Orientation. Text components belonging to one text string need to have similar orientation. As there is no universal method for the determination of the orientation of connected components, we resort to both the definition of minimum-area enclosing rectangle (MAER) [13], and R-signature [30]. MAER (green rectangles in Fig. 2(a)) can be used to determine the orientation of most characters, however, it fails with some characters like '`A`', '`r`', '`J`', *etc.*

For characters having a dominant stroke like '`r`', '`J`', '`l`', their orientation are determined as the angle corresponding the maximum of their R-signature. Fig. 2(b) shows the R-signature of character '`r`' in Fig. 2(a). The enclosing rectangles having orientations determined through the maxima of R-signatures are in blue in Fig. 2(a).

Symmetric characters like '`A`', '`x`', '`V`' also have symmetric R-signatures. Their orientation are determined as the angle that cuts their R-signature into two vectors of the same length having highest correlation. Fig. 2(c) shows the R-signature of character '`A`' in Fig. 2(a). The enclosing rectangles having orientations determined through correlation are in red in Fig. 2(a).

Let $[o_{i1}, o_{i2}, o_{i3}]$ be the three orientations of a components g_i determined by the three methods above. The difference in orientation between two components g_i, g_j is defined as:

$$O_{ij} = \min_{1 \le m,n \le 3} |o_{im} - o_{jn}|. \tag{14}$$

Thus two neighboring text components g_i and g_j need to satisfy $O_{ij} \le T_o$ to be considered as belonging to one text string. The value of T_o is determined by experience and is equal to 0.15 (radian).

Overlapping. Two neighboring text components g_i, g_j of a text string need to overlap to a certain degree along their common orientation, which is the orientation of the bisector t_{ij} of the angle formed by the two lines parallel to the orientations of g_i, g_j (see Fig. 3). Let $[a_i, b_i]$, $[a_j, b_j]$ be the orthogonal projections of g_i, g_j onto t_{ij} respectively, the degree of overlapping of two text components g_i, g_j is calculated as:

$$L_{ij} = \frac{max\{min(b_i - a_j, b_j - a_i), 0\}}{min(b_i - a_i, b_j - a_j)}. \tag{15}$$

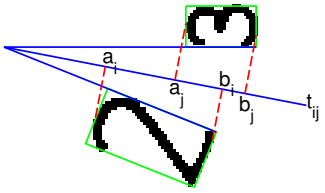


Figure 3: Determination of overlapping

The numerator of Eq. (15) is interpreted as the length of the overlapping segment. Thus two neighboring text components g_i and g_j need to satisfy $L_{ij} \ge T_l$ to be considered as belonging to one text string. The value of T_l is determined by experience and is equal to 0.75.

6. EXPERIMENTAL RESULTS

To demonstrate the efficiency of the proposed method, experiments have been carried out on the same dataset used by Tombre *et al.* [32] containing five graphical document images of different types as shown in the first column of Fig. 4. The second column of Fig. 4 provides the corresponding grayscale images containing text components obtained by using the MCA algorithm. It is clearly shown that

the MCA algorithm using undecimated wavelet and curvelet transforms as the two overcomplete dictionaries cannot totally separate text and graphics components, some parts of graphics which have local morphological characteristics like those of text still remain in text images. The third column of Fig. 4 gives the binarized images of those in the second column after removing small components (composed of less than 50 pixels). The obtained results in the third column of Fig. 4 demonstrate that text/graphics separation using the MCA algorithm overcomes the touching problem between text and graphics and is invariant to different font styles, sizes, and orientations.

A quantitative evaluation of the method is also given in Table 1. The measure used for evaluation is the recall rate of text components. The column **Nb. ch.** indicates the number of characters existing in the input images. Results of the previous benchmark [32] are given in the column **Tombre *et al.*** and the results of the proposed method are in the column **Our method**. A sharp increase in the recall rate has been achieved by using the MCA algorithm.

Image	Nb. ch.	Tombre *et al.*	Our method
1	53	49 (92.4%)	53 (100%)
2	78	59 (75.6%)	62 (79.5%)
3	78	68 (87.2%)	75 (96.2%)
4	106	92 (86.8%)	104 (98.1%)
5	21	1 (4.8%)	21 (100%)

Table 1: Performance evaluation

The technique to group text components in straight fonts into text strings has also been evaluated on the three input images, one of which is from the dataset used by Tombre *et al.* [32], showed in the first row of Fig. 5. The grouped text strings are given in the three corresponding images in the second row. Most of the text strings containing different characters and numbers of different orientations have been successfully grouped using heuristic rules proposed in Section 5. The only exception is the string `PTT(0.60)` in Image 6. The reasons for this are the connection between '`6`' and '`0`' and the embedment of '`)`' in a line.

It should be noted that in Fig. 5(d) the enclosing rectangles of text strings are not drawn on Fig. 1(e), but on Fig. 1(d). The purpose of doing this is to retrieve all the small text components like '·', ':' that have been removed previously lying inside these enclosing rectangles. This guarantees a successful extraction of all text components.

7. CONCLUSION

A novel text extraction method from graphical document images is presented in this paper using a sparse representation framework. Two discriminative dictionaries based on undecimated wavelet transform and curvelet transform are used to represent text and graphics components. Morphological Component Analysis is employed for the promotion of sparse representation of text and graphics components in these two dictionaries. The proposed method has high recall rate of text components, overcomes the problem of text/graphics touching, and outperforms the previous benchmark. Moreover, a new technique to group text components in straight fonts into text strings has proved to be efficient.

8. REFERENCES

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. on Signal Processing*, 54(11):4311–4322, 2006.

[2] H. Barlow. Unsupervised learning. *Neural Computation*, 1(3):295–311, 1989.

[3] E. J. Candès and D. L. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. *Comm. Pure Appl. Math.*, 57(2):219–266, 2002.

[4] R. Cao and C. L. Tan. Text/graphics separation in maps. In D. Blostein and Y.-B. Kwon, editors, *GREC*, volume 2390 of *LNCS*, pages 167–177. Springer, 2001.

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

[6] G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *J. Constr. Approx.*, 13(1):57–98, 1997.

[7] O. Deforges and D. Barba. A robust and multiscale document image segmentation for block line/text line structures extraction. In *Proceedings of the 12th ICPR*, volume 2, pages 306–310, 1994.

[8] D. L. Donoho. For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution. *Comm. Pure Appl. Math.*, 59(7):797–829, 2006.

[9] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[10] M. Elad, J. Starck, P. Querre, and D. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Appl. Comput. Harmon. Anal.*, 19(3):340–358, 2005.

[11] M. Fadili, J.-L. Starck, and F. Murtagh. Inpainting and zooming using sparse representations. *Computer Journal*, 52(1):64–79, 2009.

[12] L. A. Fletcher and R. Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(6):910–918, 1988.

[13] H. Freeman and R. Shapira. Determining the minimum-area encasing rectangle for an arbitrary closed curve. *Communications of the ACM*, 18(7):409–413, 1975.

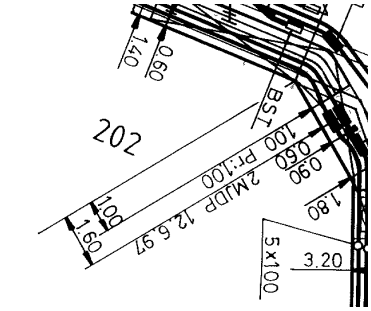
[14] J. Gloger. Use of the Hough transform to separate merged text/graphics in forms. In *Proceedings of the 11th ICPR*, volume 1, pages 268–271, 1992.

[15] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 2nd edition, 2001.

[16] T. V. Hoang, S. Tabbone, and N.-Y. Pham. Extraction of Nom text regions from stele images using area Voronoi diagram. In *Proceedings of the 10th ICDAR*, pages 921–925, 2009.

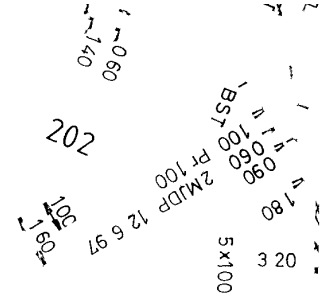
[17] C. Jutten and J. Herault. Blind separation of sources, part 1: an adaptive algorithm based on neuromimetic architecture. *Signal Process.*, 24(1):1–10, 1991.

[18] C. P. Lai and R. Kasturi. Detection of dimension sets

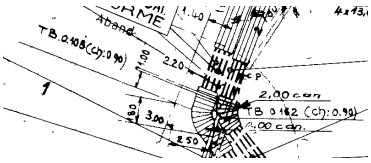


(a) Image 1: Original image

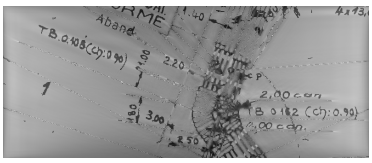
(b) Image 1: Text image



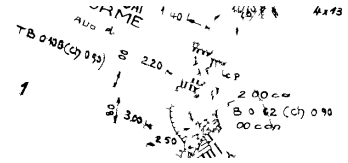
(c) Image 1: Binarized text image



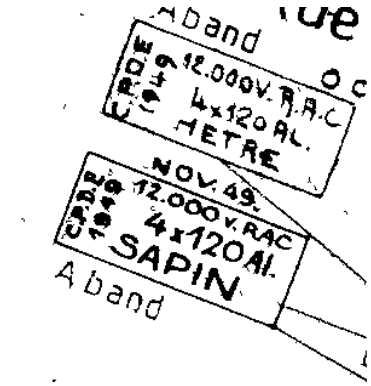
(d) Image 2: Original image



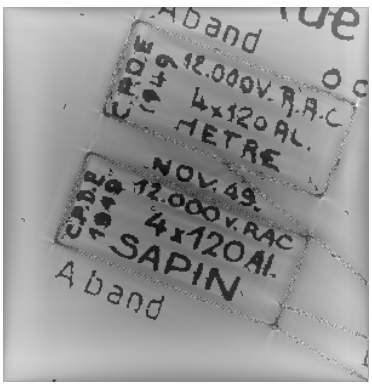
(e) Image 2: Text image



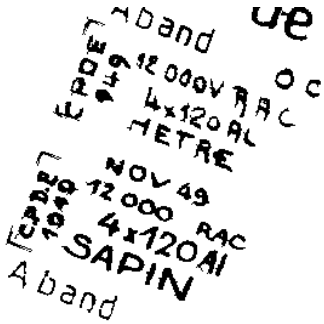
(f) Image 2: Binarized text image



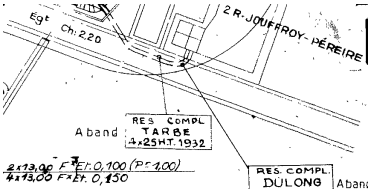
(g) Image 3: Original image



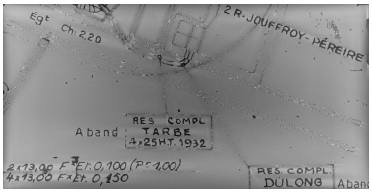
(h) Image 3: Text image



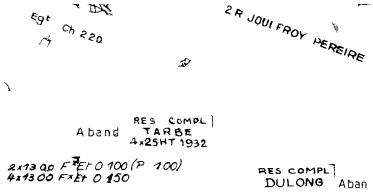
(i) Image 3: Binarized text image



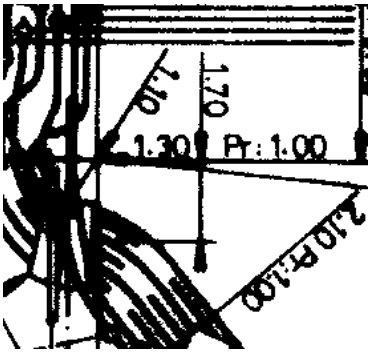
(j) Image 4: Original image



(k) Image 4: Text image

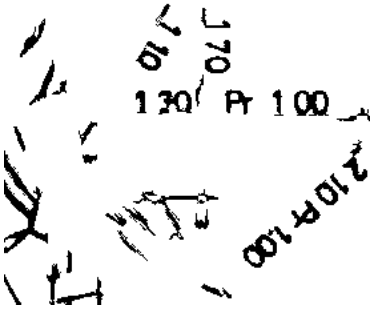


(l) Image 4: Binarized text image



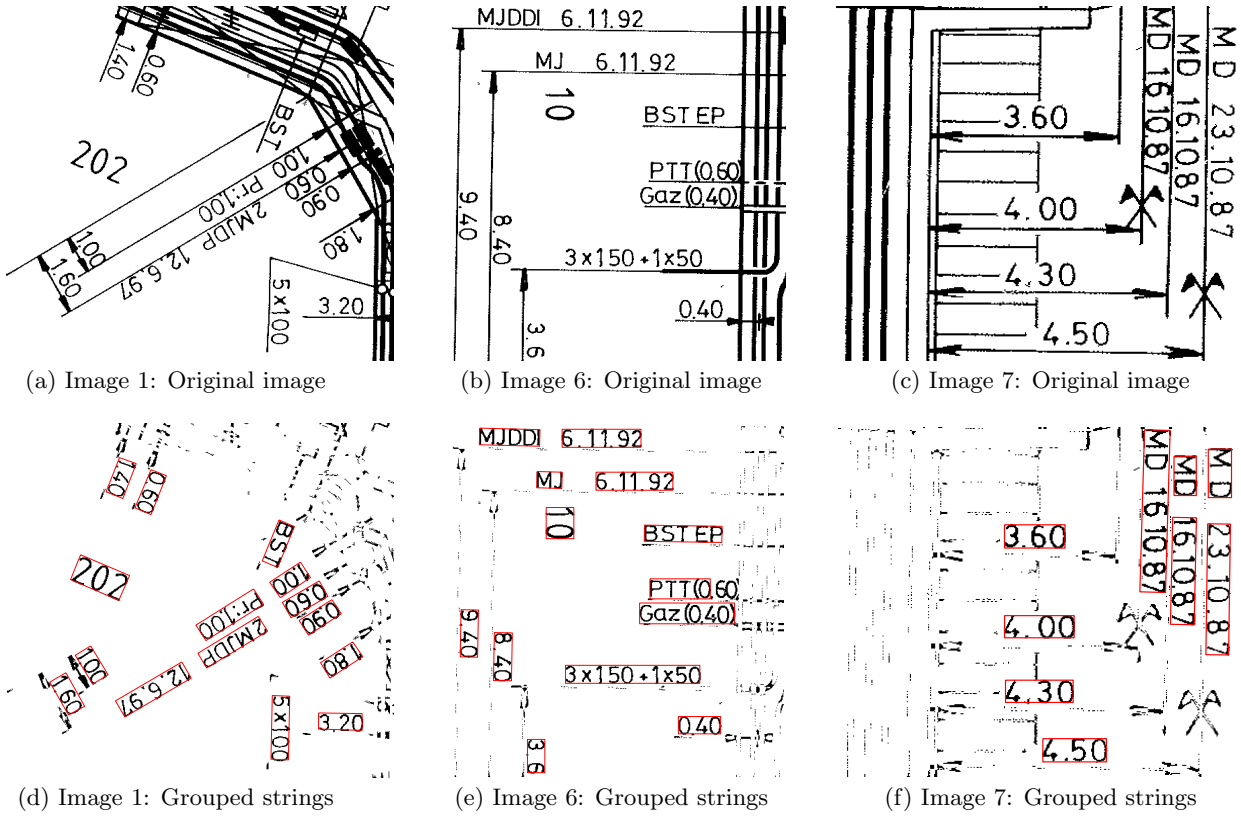
(m) Image 5: Original image

(n) Image 5: Text image



(o) Image 5: Binarized text image

Figure 4: Experimental results on text/graphics separation



(a) Image 1: Original image (b) Image 6: Original image (c) Image 7: Original image

(d) Image 1: Grouped strings (e) Image 6: Grouped strings (f) Image 7: Grouped strings

Figure 5: Experimental results on grouping text strings

in engineering drawings. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(8):848–855, 1994.

[19] D. X. Le, G. R. Thoma, and H. Wechsler. Classification of binary document images into textual or nontextual data blocks using neural network models. *Machine Vision and Applications*, 8(5):289–304, 1995.

[20] Z. Lu. Detection of text regions from digital engineering drawings. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(4):431–439, 1998.

[21] H. Luo and R. Kasturi. Improved directional morphological operations for separation of characters from maps/graphics. In K. Tombre and A. K. Chhabra, editors, *GREC*, volume 1389 of *LNCS*, pages 35–47. Springer, 1997.

[22] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[23] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision Research*, 37(23):3311–3325, 1998.

[24] W. Pan, T. Bui, and C. Suen. Text segmentation from complex background using sparse representations. In *Proceedings of the 9th ICDAR*, pages 412–416, 2007.

[25] S. Sardy, A. G. Bruce, and P. Tseng. Block coordinate relaxation methods for nonparametric wavelet denoising. *J. of Comput. and Graph. Stat.*, 9(2):361–379, 2000.

[26] M. Shensa. The discrete wavelet transform: wedding the à trous and Mallat algorithms. *IEEE Trans. on Signal Processing*, 40(10):2464–2482, 1992.

[27] J. Song, F. Su, C.-L. Tai, and S. Cai. An object-oriented progressive-simplification-based vectorization system for engineering drawings: model, algorithm, and performance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(8):1048–1060, 2002.

[28] J.-L. Starck, M. Elad, and D. L. Donoho. Image decomposition via the combination of sparse representations and a variational approach. *IEEE Trans. on Image Processing*, 14(10):1570–1582, 2005.

[29] F. Su, T. Lu, R. Yang, S. Cai, and Y. Yang. A character segmentation method for engineering drawings based on holistic and contextual constraints. In *Proceedings of the 8th GREC*, pages 280–287, 2009.

[30] S. Tabbone, L. Wendling, and J.-P. Salmon. A new shape descriptor defined on the Radon transform. *Comput. Vis. Image Underst.*, 102(1):42–51, 2006.

[31] C. L. Tan and P. O. Ng. Text extraction using pyramid. *Pattern Recognition*, 31(1):63–72, 1998.

[32] K. Tombre, S. Tabbone, L. Pélissier, B. Lamiroy, and P. Dosch. Text/graphics separation revisited. In D. P. Lopresti, J. Hu, and R. S. Kashi, editors, *DAS*, volume 2423 of *LNCS*, pages 200–211. Springer, 2002.

[33] F. Wahl, K. Wong, and R. Casey. Block segmentation and text extraction in mixed text/image documents. *Computer Graphics and Image Processing*, 20(4):375–390, 1982.