



**HAL**  
open science

# Sparsity-based edge noise removal from bilevel graphical document images

Thai V. Hoang, Elisa H. Barney Smith, Salvatore Tabbone

► **To cite this version:**

Thai V. Hoang, Elisa H. Barney Smith, Salvatore Tabbone. Sparsity-based edge noise removal from bilevel graphical document images. *International Journal on Document Analysis and Recognition*, 2014, 17 (2), pp.161-179. 10.1007/s10032-013-0213-4 . hal-00852418

**HAL Id: hal-00852418**

**<https://inria.hal.science/hal-00852418v1>**

Submitted on 21 Aug 2013

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

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Sparsity-based edge noise removal from bilevel graphical document images

Thai V. Hoang · Elisa H. Barney Smith · Salvatore Tabbone

**Abstract** This paper presents a new method to remove edge noise from graphical document images using geometrical regularities of the graphics contours that exist in the images. Denoising is understood as a recovery problem and is accomplished by employing a sparse representation framework in the form of a basis pursuit denoising algorithm. Directional information of the graphics contours is encoded by atoms in an overcomplete dictionary which is designed to match the input data. The optimal precision parameter used in this framework is shown to have a linear relationship with the level of the noise that exists in the image. Experimental results show the superiority of the proposed method over existing ones in terms of image recovery and contour raggedness.

**Keywords** Image degradation model, noise spread, bilevel image denoising, sparse representation, dictionary learning, directional denoising

## 1 Introduction

The scanning and binarization processes that produce binary document images introduce noise that is concentrated on the edges of image objects [4]. This edge noise causes difficulties in document reading and interpretation. It also has influence on later steps in the chain of

automatic document processing. For example, it could affect feature measurement in recognition, reduce image redundancy in compression, and distort skeletons in vectorization. For the accurate analysis and recognition of document images, edge noise needs to be removed.

Removing noise from images could be considered as an inverse problem where “original” images are reconstructed from available noisy images. Like many other inverse problems of similar nature, such as signal reconstruction from noisy and sampled measurements, if there exists no clue about the original image other than its noisy version and no information about the process that introduces noise to the image, then noise removal is an ill-posed or ill-conditioned problem that has many trivial solutions. For example, any image of the same size as the noisy image could be used as a solution to this problem. In order to reduce the set of solutions so that it contains acceptable ones, existing works usually rely on assumptions and noise models.

Assumptions are commonly used in the literature to recast the original inverse problem in various discrete forms, each requires a specific tool to find solutions. Usually, the original image is assumed to be “similar” to its noisy version and the geometrical regularities are “well represented” in the original image, in conformity with natural patterns. For example, similarity could be represented by means of the distance between the two points that represent the original and noisy images in a high-dimensional space, where the number of dimensions is equal to the number of pixels in each image. Since a small distance does not always guarantee a high visual quality of the denoised image, regularization is often used to prevent over-fitting and also to introduce other assumptions. On the other hand, mathematical modelling can be used to characterize the additive noise

---

Thai V. Hoang  
Inria Nancy – Grand-Est, 54600 Villers-lès-Nancy, France  
E-mail: vanthai.hoang@inria.fr, Tel: +33 3 54 95 85 60

Elisa H. Barney Smith  
Boise State University, Boise, Idaho, 83725, USA  
E-mail: ebarneysmith@boisestate.edu

Salvatore Tabbone  
Université de Lorraine, LORIA, UMR 7503, 54506  
Vandoeuvre-lès-Nancy, France, E-mail: tabbone@loria.fr

that results from a number of degradation factors in the image formation process. Intuitively, different levels of noise demand different “levels of denoising”, or different levels of regularization. Thus, knowledge about the level of additive noise may be useful in controlling the aforementioned regularization.

This paper revisits the denoising problem in the context of graphical document images in light of recent research in document image degradation modelling [4] and recent developments in signal representational theories [27] in order to achieve the two preferred performance criteria of image recovery and contour smoothness. It is a thorough extension of a preliminary work presented in [28] and a continuation of a previous work that uses sparsity for text/graphics separation [29]. The remainder of this paper is organized as follows. Section 2 gives some background on edge noise modelling, followed by a detailed review of related works in Section 3. The proposed edge noise removal algorithm using sparse representation is presented in Section 4. Experimental results and discussions are given in Section 5, and conclusions are drawn in Section 6.

## 2 Image degradation model

A scanner model which is based on the physics in the document acquisition process provides a theoretical platform for the analysis of that process. The scanner model that is described below is a portion of the model presented in [3] and is schematically described in Fig. 1. It is assumed in this model that when a spatially continuous image  $o$  is converted to digital form using either a digital camera or a document scanner, the value of each pixel in the scanned image before quantization,  $s[i, j]$ , depends on the light collected at the corresponding discrete sensor. This collected light in turn depends on the reflectance in the original image in the neighborhood around that sensor, that is a function of the optics and the sensors. The contribution of the source reflectance to the sensor value is usually described by a function of the distance from the sensor center, called the point spread function (PSF). Thus, the signal value that is received at each sensing element  $(i, j)$  is modeled as

$$s[i, j] = \iint \text{PSF}(x_i - u, y_j - v, w) o(u, v) dudv.$$

In order to model the noise that would occur during the acquisition process, it is generally assumed that additive white Gaussian noise (AWGN)  $n$  of standard deviation  $\sigma_{noise}$  is added to these values as

$$a[i, j] = s[i, j] + n[i, j].$$

Moreover, since document and graphical images are usually processed in bilevel, this image is then thresholded, usually with a global threshold value  $\Theta$  as

$$f[i, j] = \begin{cases} 1, & a[i, j] \geq \Theta \\ 0, & a[i, j] < \Theta. \end{cases}$$

While the AWGN is evenly distributed over the whole grayscale image, the effect of AWGN after thresholding is concentrated along the edges. The process of turning a smooth edge into a rough one and the analysis of this rough edge are discussed as follows.

### 2.1 Edge without noise

In graphical document images, the image content contains large regions of white (0) background, with foreground image features displayed in black (1). When documents are scanned in grayscale, the edges change from step functions to functions sloped in the shape of the edge spread function (ESF), which is the cumulative marginal of the PSF. The slope of the edge functions in turn causes changes in the edge locations after thresholding [5]. For a PSF parameterized by a width  $w$ , the amplitude of the ESF is

$$s(x) = \text{ESF}\left(\frac{x}{w}\right).$$

An example of an ESF is shown in Fig. 2a and an image that corresponds to this is shown in Fig. 3c. When there is no noise, the edge location after thresholding occurs at the point where the amplitude is equal to the threshold  $\Theta$  and would be at the location  $x = -\delta_c$ , where

$$\delta_c = -w \text{ESF}^{-1}(\Theta).$$

The shift  $\delta_c$  in edge location that depends on  $s$  and  $\Theta$  in Fig. 2a is depicted in Fig. 2b. This effect in images is shown in Fig. 3e.

### 2.2 Edge with noise and noise spread

When additive noise  $n$  is considered, the edge intensity after blurring  $a$  will fluctuate around  $s$  as illustrated in Fig. 2c. This variation in an image is shown in Fig. 3d. This fluctuation results in a small region near the edge of the step function in which the value of the pixel could be above or below the threshold. So, after thresholding the thresholded signal has a noisy edge that may be at any position in that region, Fig. 3f. The breadth of that region is defined as noise spread (NS) (Fig. 2d). It has

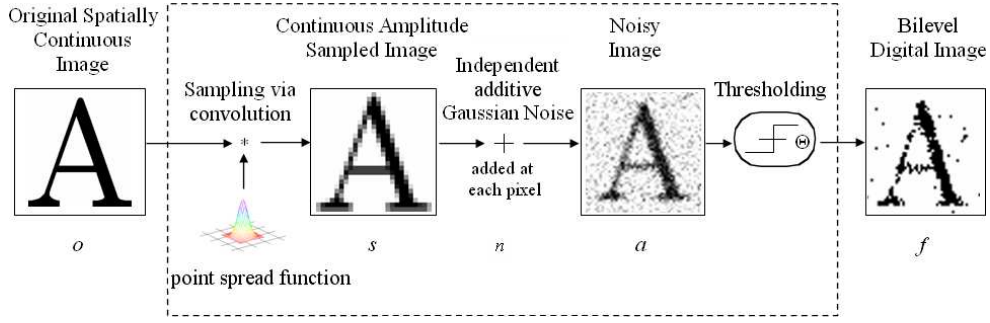


Fig. 1: The scanner model used to determine the value of the pixel  $[i, j]$  centered on each sensor element. It is modeled as a multi-stage process whose steps include convolution with a point spread function (PSF), sampling, adding noise, and thresholding.

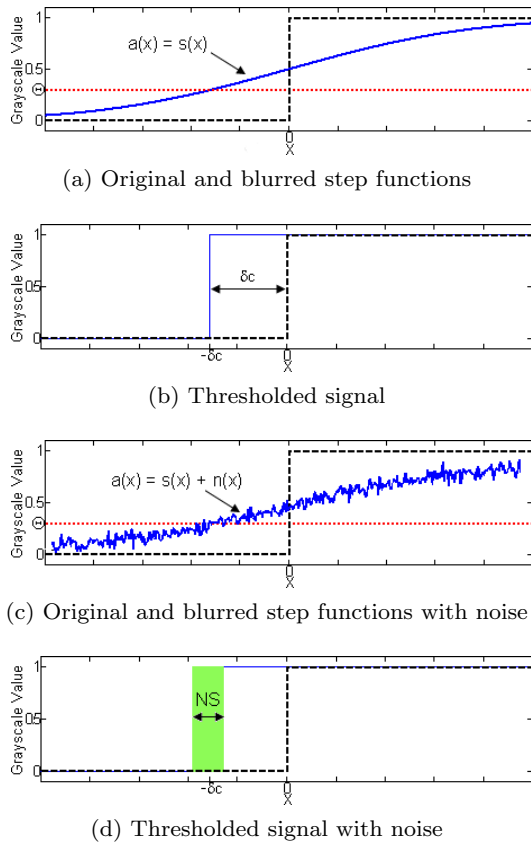


Fig. 2: The original edge is a step function with a dashed line. (a) Edge after blurring with a generic PSF of width  $w$ . (b) When no noise is added, the thresholding produces the edge shift  $\delta_c$ . (c) A blurred edge with added noise after thresholding produces (d) the uncertain boundary region, called the NS region, shown shaded.

been shown recently [36] that NS is not just dependent on the standard deviation  $\sigma_{noise}$  of AWGN, but also on the width  $w$  of the PSF, which determines the slope of the grayscale edges, and the level of the binarization

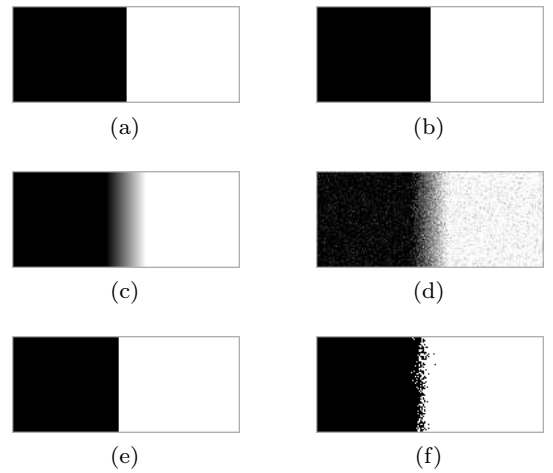


Fig. 3: Top view of representations for Fig. 2. The left column shows the original step function,  $o(x)$ ; the blurred step function,  $s(x)$ ; and the thresholded edge  $f(x)$  without noise. The right column shows the original step function,  $o(x)$ ; the blurred edge  $a(x)$  with added noise; and the noisy edge after thresholding.

threshold  $\Theta$  through the relation

$$NS = \frac{\sqrt{2\pi} \cdot \sigma_{noise} \cdot w}{LSF(ESF^{-1}(\Theta))},$$

where LSF is the line spread function, or 1D PSF.

Some examples of edges with noise are shown in Fig. 4. The standard deviation  $\sigma_{noise}$  of the additive noise is kept fixed in the three images in Figs. 4a–4c. Images with a common  $\sigma_{noise}$  are conventionally thought of as having the same noise level. However, it can be seen that these three images have edges with distinctly different amounts of perceptual noise. Even with a constant value of  $\sigma_{noise}$ , an increase in NS makes the image noisier in an amount proportional to the increase in the value of NS. In Fig. 4d, a significantly

larger NS is shown and its effect on the edge can be easily observed: there are generally two rows of pixels affected by the additive noise when  $NS = 2.0$ . In this manner, NS provides a measure that can numerically quantify the amount of edge noise and also relates to the noise visually observed on bilevel images.

### 2.3 Relationship between NS and Hamming distance

The real benefit of determining the NS of a scanned object is that NS provides an effective measure of how noisy a bilevel object is. However, for two bilevel images of the same size, the Hamming distance [26], which is defined as the number of substitutions required to change one image into the other, is often used as a measure of difference between them. It is very useful and hence is usually used for the analysis of noise in bilevel images when the noise-free template is known. The formula that relates the expected Hamming distance  $E\{H\}$  to NS for straight edges was developed [36] as

$$E\{H\} = \frac{NS \cdot \rho}{\pi}, \quad (1)$$

where  $\rho$  is a constant equal to the length of the edge segment. The above equation shows that Hamming distance is directly proportional to NS, leading to the possibility of using NS as a predictor of the Hamming distance between a degraded image and the predicted noise-free image, and vice versa. In addition to the theoretical result in Eq. (1), experiments have also been carried out to validate this linear relationship between NS and the expected Hamming distance.

The ability of NS to provide a quantitative measure that also qualitatively describes the amount of noise and its linear relationship with the Hamming distance led to a potential that NS could be used as an input to a denoising algorithm that works on bilevel images, in a similar fashion to how the standard deviation of the AWGN is often used as an input to denoising algorithms that work on grayscale images.

### 3 Related bilevel denoising work

Let the noisy image  $f$  be the result of scanning and then global thresholding a noise-free input image  $f_0$  of size  $w \times h$ , it will contain edge noise of a certain NS. Denoising is viewed as an inverse problem, i.e., one needs to find an estimated image  $\hat{f}$  from  $f$  which is close to  $f_0$  and, at the same time, has some preferred properties like contour smoothness for graphical document images.

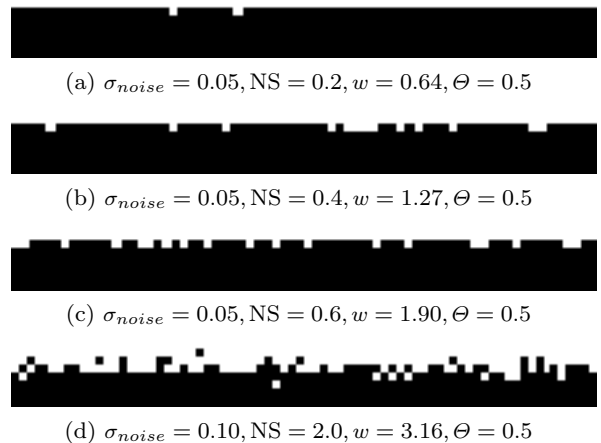


Fig. 4: Illustrations of edges with varying amounts of NS. (a)–(c) While the noise standard deviation is the same for each image, NSs are different. (d) NS qualitatively describes how many edge rows are affected.

Many methods exist for removing noise from digital images [8], each has its own properties that make it suitable for some particular situations. The aim of this section is thus not to give a long list of existing methods, but to review some ones relevant to the problem of noise removal from bilevel graphical document images based on the preferred criteria of image recovery and contour smoothness.

*Bilevel image denoising:* For bilevel document images, the most famous and frequently used denoising methods are contour smoothing using chain codes, median filtering, morphological operators, and kFill filtering. Contour smoothing based on chain codes [52] replaces a chain code sequence by a simpler sequence, usually a shorter one that corresponds to the shortest path. The simplicity in the code is enforced by minimizing the total change in the code sequence, which in turn is done by recursively replacing two consecutive changes by a single one. Even though this method can produce a smooth contour from a jagged one, it cannot be performed on an image such as the one that is shown in Fig. 5a where the noisy pixels are not only distributed along the contours, but also over the whole image region.

The main idea of the median filter [2] is to run through the image pixel by pixel, replacing each pixel’s value with the median of neighboring pixels’ values. Due to its nature, the median filter is particularly effective at removing outliers, such as “salt & pepper” noise or noise whose probability density has heavier tails than the Gaussian. Morphological operators [35] like

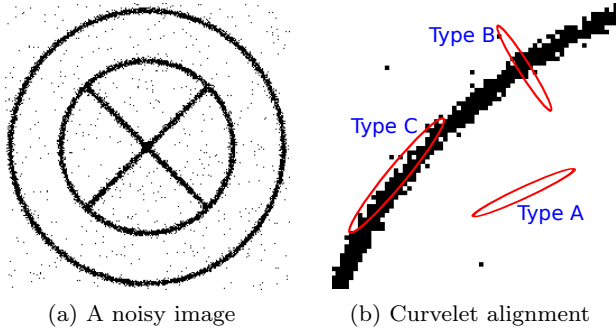


Fig. 5: Geometric illustration of directional denoising using curvelets: (a) sample noisy graphical image; (b) three types of alignment of curvelets with a noisy graphics contour, only curvelets of type C capture the graphics contour and have significant magnitude.

erosion, dilation, and their combinations opening and closing have their root in set and lattice theories. The popularity and efficiency of the simple morphological openings and closings to suppress positive and negative impulse noise have theoretical supports. The kFill algorithm [38] is designed to remove “salt & pepper” noise iteratively while maintaining the readability of text by using a filter that retains text corners. The value of the parameter  $k$  of the filter can be chosen adaptively based on text size and image resolution.

Among the aforementioned denoising methods, median filtering, morphological operators, and kFill filtering perform isotropic and geometric local smoothing and thus may not be sufficient for denoising tasks that need directional smoothing or contour preservation. These methods are known to be unable to preserve fine image details and may unintentionally remove thin lines and corners. This is because they are general-purpose denoising methods which are not designed specifically for edge noise and do not exploit the directional information in their operations. Contours denoised by these methods are jagged and sometimes shifted from their original positions. Another shortcoming of existing binary image denoising methods is that they do not take into account the information about the level of noise that exists in the binarized document images. Denoising is performed in a “blind”, non-adaptive way.

*Total variation denoising:* Noise removal by minimizing the total variation (TV) of an image  $f$ ,  $\text{TV}(f) = \int |\nabla f(\mathbf{x})| \, d\mathbf{x}$ , while preserving some fit to the original measured data was first proposed in [45]. The method is based on the principle that images with excessive and possibly spurious details have a high TV, that is the

integral of the absolute gradient of the image is high. According to this principle, the problem of removing noise from a noisy image  $f$  based on TV could be posed as the following optimization problem:

$$\hat{f} = \underset{y}{\operatorname{argmin}} \text{TV}(y) \quad \text{subject to} \quad \|y - f\|_2 \leq \epsilon_{\text{TV}}, \quad (2)$$

where the parameter  $\epsilon_{\text{TV}}$ , which is related to the estimated noise level, determines the sharpness or smoothness of the restored image. It has been proven that [47] for a general texture image this noise removal method has an edge-preserving property, which is better than simple methods such as linear smoothing or median filtering, which reduce noise, but at the same time smooth away edges. However, the edge-preserving effects of TV regularization is somewhat local; that is, the effect on one edge in the image has little or no correlation with the effect on another edge. This local property results in the inability of the TV-based denoising method to exploit the long global contours that exist in graphical document images in order to produce smooth ones.

*Anisotropic diffusion:* In image processing, anisotropic diffusion aims at suppressing image noise without removing significant parts of the image content. It is motivated by minimizing the energy functional of an image  $f$  defined by

$$E_f = \frac{1}{2} \int_{\Omega} g(\|\nabla f(x)\|^2) \, dx,$$

where  $g$  is a real-valued function. The gradient descent equation is given by

$$\frac{\partial f}{\partial t} = -\nabla E_f = \operatorname{div}(g'(\|\nabla f(x)\|^2) \nabla f).$$

Thus by letting  $c = g'$ , the anisotropic diffusion equation becomes

$$\frac{\partial f}{\partial t} = \operatorname{div}(c(x, y, t) \nabla f) = \nabla c \cdot \nabla f + c(x, y, t) \Delta f,$$

where  $\Delta$  denotes the Laplacian,  $\nabla$  denotes the gradient, and  $\operatorname{div}(\cdot)$  is the divergence operator.

In the original formulation [42], the diffusion coefficient  $c(x, y, t)$  that controls the diffusion rate was proposed to be

$$c(\|\nabla f\|) = e^{-(\|\nabla f\|/K)^2}, \quad \text{or} \quad \frac{1}{1 + \left(\frac{\|\nabla f\|}{K}\right)^2},$$

where  $K$  is a constant that controls the sensitivity to edges. The filter is in fact isotropic, but depends on the image content in the way that it approximates an impulse function close to edges and other image’s structures that need to be preserved over the different levels

of the resulting scale-space. A more general formulation allows the filter to adapt locally to be truly anisotropic near linear structures such as edges or lines. The filter has an orientation similar to that of the structure such that it is elongated along the structure and narrow across. Such a method is referred to as *coherence enhancing diffusion* [50]. As a consequence, the resulting images preserve linear structures while at the same time smoothing is made along these structures.

*Orthogonal wavelet denoising:* Wavelet-based image denoising has been used widely and its success is due to the tendency of images to have a compact representation in the wavelet transform domain [43]. The efficiency of image approximation based on a small subset of wavelets also led to the adoption of the wavelet transform in JPEG-2000 image compression and coding systems. In denoising problems, it is legitimate to assume that only a few large wavelet coefficients contain information about the underlying images while small coefficients are attributed to the noise. Thus, the common procedure is to first apply the discrete wavelet transform (DWT) (analysis operator  $\mathbf{T}$ ) to the noisy image  $f$ , then to use a nonlinear estimation rule  $\mathcal{O}$  to the transform coefficients, and finally to compute the inverse DWT (synthesis operator  $\mathbf{T}^T = \mathbf{T}^{-1}$ ) to get an estimate  $\hat{f}$  of the noisy image  $f$ . This procedure can be described symbolically as

$$\hat{f} = \mathbf{T}^T \mathcal{O}(\mathbf{T}f). \quad (3)$$

This approach has already proven to be very successful on both practical and theoretical sides [31]. Many thresholding or shrinkage rules were proposed for the operator  $\mathcal{O}$  with hard-thresholding and soft-thresholding certainly being the most well-known. For the 1D variable  $t$ , these thresholding operators are defined as follows.

- *Hard-thresholding* [49] consists of setting to zero all coefficients whose magnitudes are less than a threshold (Fig. 6a):

$$\mathcal{O}(t, \lambda) = \begin{cases} t, & |t| \geq \sqrt{2\lambda} \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

- *Soft-thresholding* [20] is defined as the kill-or-shrink rule with the coefficients above a threshold are shrunk toward the origin and those with a magnitude smaller than that threshold are set to zero (Fig. 6b):

$$\mathcal{O}(t, \lambda) = \text{sign}(t)(|t| - \lambda)^+, \quad (5)$$

where  $(\cdot)^+ = \max(\cdot, 0)$ .

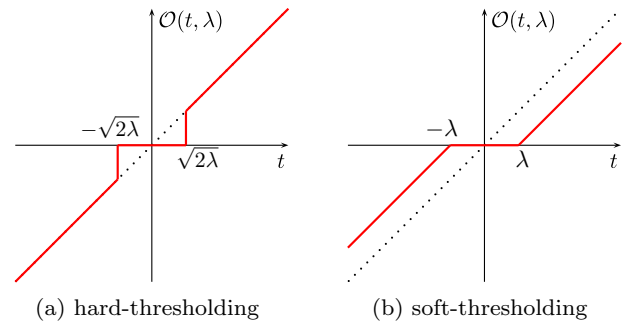


Fig. 6: Illustrations of the hard-thresholding and soft-thresholding operators.  $\mathcal{O}$  is a function of the input  $t$  for a fixed  $\lambda$ . The black dotted curve is the function  $\mathcal{O}(\cdot, 0)$  (no regularization), whereas the red plain curve corresponds to  $\mathcal{O}(\cdot, \lambda)$ .

*Noise modelling:* It can be easily seen that most of the denoising methods mentioned above are parametric and they require knowledge of noise to set their parameters in order to have good performance. For example, TV denoising needs to set the value for  $\epsilon_{\text{TV}}$ ; multiscale denoising has to do the same for  $\lambda$ . The knowledge of noise is thus a crucial factor in applying noise removal methods. In practical situations, the noise level is usually obtained from image noise models [7], which are built from the knowledge of noise generation processes or from some measured values. For the aforementioned edge noise which can be measured quantitatively, the availability of NS could pave the way for the design of a new more efficient method to remove edge noise.

The following section describes a new parametric method for edge noise removal in bilevel graphical document images that exploits the directional information of graphics contours. Information about the level of edge noise represented by NS is used as an input to the denoising process in order to have better performance. Directional denoising is facilitated by using a sparse representation framework. This is done by promoting the sparse representation of graphical document images in an overcomplete dictionary using a basis pursuit denoising algorithm with the dictionary defined from curvelets or learned from data. The images reconstructed from their sparse representations are grayscale ones, which can be simply thresholded to obtain the final bilevel denoised images.

#### 4 Sparsity-based edge noise removal

Let  $\mathbf{x}_0$ ,  $\mathbf{x}$ , and  $\hat{\mathbf{x}} \in \mathbb{R}^p$  ( $p = wh$ ) be the vectors generated by stacking the columns of  $f_0$ ,  $f$ , and  $\hat{f}$  respectively

then  $\mathbf{x} = \mathbf{x}_0 + \mathbf{z}$ , where  $\mathbf{z} \in \mathbb{R}^s$  stands for the unknown additive edge noise. This section is devoted to the finding of  $\hat{\mathbf{x}}$  from  $\mathbf{x}$  by using the recent idea of directional representation in order to achieve the two preferred criteria of image recovery and contour smoothness.

#### 4.1 Multiscale directional denoising

Although applications of wavelets in image processing have become increasingly popular, traditional wavelets represent well only point singularities since they ignore the geometric properties of structures and do not exploit the regularity of edges. Images denoised by using traditional wavelets usually have unfavorable blocky artifacts. Therefore, wavelet-based denoising becomes inefficient for geometric line-like features and surface singularities.

To overcome the missing directional selectivity of conventional 2D DWTs, there have been several developments of wavelet frames in recent years. Steerable wavelets [24], Gabor wavelets [32], brushlets [37], beamlets [19], ridgelets [16], curvelets [9], contourlets [17], shearlets [25], wave atoms [15], surflets [11] were proposed independently with the goal of better representation of directional features in images. Among these X-lets, curvelets have the highest publicity and have found applications in several domains [33]. In the 2D case, the curvelet transform allows an almost optimal sparse representation of objects with singularities along smooth curves. For a smooth object  $f$  with piecewise  $C^2$  singularities, the best  $N$ -term approximation  $\hat{f}_N$ , that is a linear combination of only  $N$  elements of the curvelet frame, obeys  $\|f - \hat{f}_N\|_2 \leq CN^{-2}(\log N)^3$ , while for wavelets the decay rate is only  $N^{-1}$ .

The curvelet transform has a property that the coefficients of those curvelets whose essential support does not overlap with or overlaps with, but are not tangent to an edge are small and negligible. For example, in Fig. 5b, coefficients of curvelets of types A and B are negligible; most of the energy of the graphics is localized in just a few coefficients of type C. In other words, the curvelet transform produces a sparse representation of objects, most of the energy of the objects is localized in just a few coefficients of curvelets which overlap and are nearly tangent to the object contours. Based on this property, the application of the curvelet transform for image denoising is straightforward; it could simply be done by hard-thresholding of curvelet coefficients as in Eq. (4) [46]. The images reconstructed by curvelets exhibit higher perceptual quality than those

from wavelets, have visually sharper and, in particular, higher quality recovery of edges and of linear and curvilinear features.

#### 4.2 Thresholding as a minimization problem

Suppose that denoising is performed by using Eq. (3) where  $\mathbf{T}$  is orthonormal like in the case of the DWT. Let's investigate the following optimization problem to obtain an estimated image  $\hat{\mathbf{x}} = \mathbf{T}^T \hat{\boldsymbol{\alpha}}$ :

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_q^q, \quad (6)$$

where  $\mathbf{D} = \mathbf{T}^T = \mathbf{T}^{-1}$ . Due to the unitarity of  $\mathbf{T}$ :

$$\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 = \|\mathbf{x} - \mathbf{T}^T \boldsymbol{\alpha}\|_2^2 = \|\mathbf{T}\mathbf{x} - \boldsymbol{\alpha}\|_2^2,$$

and thus the optimization translates into

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{\beta} - \boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_q^q, \quad (7)$$

where  $\boldsymbol{\beta} = \mathbf{T}\mathbf{x}$  is the transform of the noisy image  $\mathbf{x}$ . As both terms  $\|\boldsymbol{\beta} - \boldsymbol{\alpha}\|_2^2$  and  $\|\boldsymbol{\alpha}\|_q^q$  are separable, the optimization problem in Eq. (7) decouples into a set of  $n$  independent scalar problems of the form

$$\hat{\alpha}_i = \underset{\alpha_i}{\operatorname{argmin}} \frac{1}{2} \|\beta_i - \alpha_i\|_2^2 + \lambda \|\alpha_i\|_q^q, \quad i = 1, \dots, n. \quad (8)$$

It is not difficult to prove that the unique closed-form solution of this problem in the two notable cases  $q = 0$  and  $q = 1$  are actually the two thresholding operators defined in Eqs. (4) and (5) respectively with  $\alpha_i = \mathcal{O}(\beta_i, \lambda)$ . Thus, for the case of orthonormal transforms, thresholding-based denoising could be viewed as solving an optimization of the form in Eq. (6) with  $q = 0$  for hard-thresholding and  $q = 1$  for soft-thresholding.

#### 4.3 Basis pursuit denoising (BPDN)

The thresholding operators defined in Eqs. (4) and (5) are the exact solutions to the optimization problem in Eq. (6) for the two cases  $q = 0$  and  $q = 1$  only if  $\mathbf{D}$  is orthonormal. When a redundant transform like the curvelet transform is used, the corresponding overcomplete dictionary  $\mathbf{D}$  has more columns than rows and is not unitary ( $\mathbf{D}\mathbf{D}^T = \mathbf{I}$  but  $\mathbf{D}^T\mathbf{D} \neq \mathbf{I}$  where  $\mathbf{I}$  is the identity matrix). The problem in Eq. (6) thus does not have a simple and closed-form solution, even in the two notable cases  $q = 0$  and  $q = 1$ . This is because the presence of  $\mathbf{D}$  destroys the separability that allows the solving of Eq. (8) instead of Eq. (7).



However, for the graphical document image denoising problem, the formulation in Eq. (6) is still adopted with the overcomplete dictionary  $\mathbf{D}$  being defined as the synthesis operator of the curvelet transform in order to obtain smooth graphical contours. Moreover, to ease the investigation of the dependence of the framework's parameter on the noise level, the optimization problem is written as

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \|\boldsymbol{\alpha}\|_1 \quad \text{subject to} \quad \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2 \leq \epsilon, \quad (9)$$

where  $\epsilon$  is the precision parameter depending on  $\mathbf{z}$ . Note that the above formulation is a slightly modified version of the BPDN problem in [12] where the squared Euclidean norm is replaced by the Euclidean norm. This modification in fact does not change the nature of the problem as the value of  $\epsilon$  could be changed accordingly:

$$\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2 \leq \epsilon \quad \Leftrightarrow \quad \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 \leq \epsilon^2.$$

The BPDN problem has been well-studied by optimization specialists and there are many practical methods for solving it. For large-scale applications, the following special purpose optimizers are frequently used: iterative reweighted least squares [13], iterative shrinkage-thresholding [6], least angle regression [21]. Interestingly, it has been shown that simple shrinkage could be interpreted as the first iteration of an algorithm that solves BPDN [23]. By solving the problem in Eq. (9), the estimated image can be obtained from the sparse reconstruction as  $\hat{\mathbf{x}} = \mathbf{D}\hat{\boldsymbol{\alpha}}$ .  $\hat{\mathbf{x}}$  is of course in grayscale due to reconstruction from curvelets and is then converted to bilevel by a simple thresholding operation  $\tilde{\mathbf{x}} = \mathcal{T}(\hat{\mathbf{x}})$  where  $\mathcal{T}$  is the thresholding operator.

In the above BPDN problem, the  $l_1$ -norm is used instead of a more general  $l_q$ -norm to avoid the NP-hard problem [14] when  $q = 0$ , non-convexity when  $q < 1$ , non-sparse and over-fitting solutions when  $q > 1$ . In addition, a sparse solution, which guarantees directional denoising by curvelets, is still obtained if the solution of the following  $l_0$ -norm optimization problem is sufficiently sparse [18]:

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \|\boldsymbol{\alpha}\|_0 \quad \text{subject to} \quad \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2 \leq \epsilon.$$

As the overcomplete dictionary  $\mathbf{D}$  is constructed from curvelets and the images to be processed contain mainly graphical contours, this requirement is easily satisfied. Illustration of the distribution of the magnitude of the 5000 largest curvelet coefficients of the image in Fig. 5a obtained from the curvelet transform and BPDN is given in Fig. 7. It is observed that BPDN results in a sparse representation; many elements of  $\hat{\boldsymbol{\alpha}}$  have almost

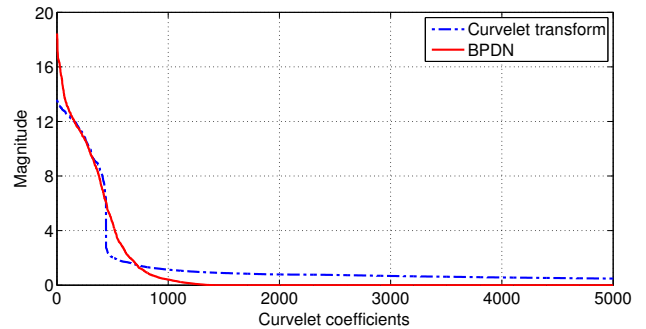


Fig. 7: The distribution of the magnitude of the 5000 largest coefficients of the noisy image in Fig. 5a obtained from curvelet transform and BPDN with  $\epsilon = 48$ . Small-valued coefficients resulting from curvelet transform are zeroed out by BPDN.

zero value. The sparsity in  $\hat{\boldsymbol{\alpha}}$  is, in some respects, better than that in the coefficients  $\mathbf{D}^T\mathbf{x}$  resulting from the curvelet transform. In addition, the shape of the distribution of  $\hat{\boldsymbol{\alpha}}$  resembles that of a Laplacian distribution and this agrees with the Bayesian formulation of sparse coding [51].

It should also be noted that the problem in Eq. (9) and that in Eq. (2) are similar, they are both minimization problems with a fidelity constraint. The main difference between them is that TV denoising pursues an estimation that is sparse in the spatial domain (sparse gradient) whereas BPDN denoising has a desire for sparsity in the transform domain. Fusing TV with BPDN amounts to solving [10]

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \operatorname{TV}(\boldsymbol{\alpha}) \quad \text{subject to} \quad \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2 \leq \epsilon.$$

However, as the gradient of  $\boldsymbol{\alpha}$  is not clearly defined, imposing piece-wise smoothness in  $\boldsymbol{\alpha}$  by TV regularization in this way usually produces images with non-deterministic artifacts.

#### 4.4 The precision parameter $\epsilon$

The BPDN problem in Eq. (9) has a non-negative precision parameter  $\epsilon$  that describes the desired fidelity of the reconstructed  $\hat{\mathbf{x}}$  to the noisy  $\mathbf{x}$ . It is the only parameter, besides the selected dictionary  $\mathbf{D}$ , that controls the quality of denoised images. It is straightforward that when  $\epsilon = 0$ , the BPDN problem reduces to a simple curvelet transform:

$$\mathbf{x} = \mathbf{D}\hat{\boldsymbol{\alpha}} \quad \longrightarrow \quad \hat{\boldsymbol{\alpha}} = \mathbf{D}^T\mathbf{x}$$

and one easily has  $\mathbf{x} = \hat{\mathbf{x}} = \tilde{\mathbf{x}}$ . As the value of  $\epsilon$  increases, the measure of sparsity  $\|\hat{\boldsymbol{\alpha}}\|_1$  of the solution

$\hat{\alpha}$  must monotonically decrease, since the feasible set of solutions  $\mathcal{S} = \{\alpha : \|\mathbf{x} - \mathbf{D}\alpha\|_2 \leq \epsilon\}$  gets wider, taking the feasible set of solutions of a smaller  $\epsilon$  as a subset:

$$\epsilon_1 < \epsilon_2 \quad \longrightarrow \quad \mathcal{S}_1 \subset \mathcal{S}_2.$$

A sparser solution means a better alignment of curvelets with graphics contours, which consequently increases denoising performance. However, when  $\epsilon$  has a reasonably large value, the solution  $\hat{\alpha}$  of the BPDN problem may be overly sparse in terms of the  $\ell_1$ -norm and the estimated image  $\hat{\mathbf{x}}$  gets overly blurred. In addition, due to the thresholding operation to get the binary image  $\tilde{\mathbf{x}}$  from  $\hat{\mathbf{x}}$ , deformation in  $\tilde{\mathbf{x}}$  will appear. Illustration of the influence of the value of  $\epsilon$  on the estimated images is given in Fig. 8 using the noisy image in Fig. 5a at  $\epsilon = 30, 40, 50, 60$ . It can be seen that for both thresholding operations using a fixed threshold of 0.5 or using Otsu's threshold [40]:

- A small value of  $\epsilon = 30$  results in insufficient blurring in the estimated images. The binarized images still have noise along the contours.
- A large value of  $\epsilon = 60$  causes over blurring in the estimated images. Deformation can be observed in the binarized images.

The selected value of  $\epsilon$  should depend on the level of noise that exists in the images. In the literature, there exists no work that discusses in detail the dependence of  $\epsilon$  on an image's noise level. For zero-mean white and homogeneous Gaussian noise with a known standard-deviation  $\sigma$ , the value of  $\epsilon$  is usually chosen as  $c n \sigma^2$ , with  $0.5 \leq c \leq 1.5$  [22, Chapter 14]. For graphical document images, the theory of edge noise presented in Section 2 sheds light on this problem by the established linear relationship between NS and the expected Hamming distance as given in Eq. (1). It is thus fair to conjecture that the relation  $\epsilon(\text{NS})$  should also be linear and is of the form  $\epsilon = k\text{NS}$ . This is because  $\|\mathbf{x} - \mathcal{T}(\mathbf{D}\alpha)\|_2$  is essentially the Hamming distance between the binary denoised image  $\tilde{\mathbf{x}} = \mathcal{T}(\mathbf{D}\alpha)$  and its corresponding noisy one  $\mathbf{x}$ . Experimental evidence of the linear relationship between  $\epsilon$  and NS will be shown in the experimental section (Section 5). The remainder of this paper uses a fixed level of 0.5 for the thresholding operation since it gives better bilevel images than using Otsu's method.

#### 4.5 BPDN with a learned dictionary

The above formulations use the synthesis operator of the curvelet transform as the overcomplete dictionary

$\mathbf{D}$ . Thus  $\mathbf{D}$  is fixed regardless of the input image  $\mathbf{x}$  that needs to be represented. Allowing  $\mathbf{D}$  to change leads to the problem of dictionary learning, where  $\mathbf{D}$  is learned to optimally adapt to a certain class of images. Dictionary learning for natural images under the sparsity assumption [39] aims at maximizing the likelihood that natural images have efficient, sparse representations in an overcomplete dictionary. Mathematically speaking, for a given image  $\mathbf{x}$ , the goal of learning is to find an overcomplete dictionary  $\mathbf{D}^*$  such that

$$\mathbf{D}^* = \underset{\mathbf{D}}{\operatorname{argmax}} \log \int_{\alpha} p(\mathbf{x}|\mathbf{D}, \alpha) p(\alpha) d\alpha. \quad (10)$$

In order to solve the above difficult optimization problem, two main assumptions were introduced. First, the coefficients  $\alpha_i$  are independent and each has a Laplacian distribution, which nicely fits the probability distribution of  $\alpha_i$  when the decomposition is sparse. The second assumption is that the reconstruction error  $\mathbf{x} - \mathbf{D}\alpha$  can be modeled as the additive white Gaussian noise. Under these two assumptions, the problem in Eq. (10) transforms into the minimization problem:

$$\mathbf{D}^* = \underset{\mathbf{D}, \alpha}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{x} - \mathbf{D}\alpha\|_2^2 + \lambda \|\alpha\|_1.$$

When a set of  $n$  images  $\mathbf{X} = [\mathbf{x}^1, \dots, \mathbf{x}^n]$  is used to learn the dictionary  $\mathbf{D}$ , the above problem becomes

$$\mathbf{D}^* = \underset{\mathbf{D}, \mathbf{A}}{\operatorname{min}} \frac{1}{2} \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_2^2 + \sum_{i=1}^n \lambda \|\alpha^i\|_1, \quad (11)$$

where  $\mathbf{A} = [\alpha^1, \dots, \alpha^n]$ . This problem could also be viewed as a matrix factorization problem using  $\ell_1$  regularization where the matrix  $\mathbf{X}$  is factored into the two matrices  $\mathbf{D}$  and  $\mathbf{A}$ . In the literature, this problem is usually solved by alternating between two steps:

- *Sparse coding*:  $\mathbf{D}$  is kept constant, the energy function is minimized with respect to  $\mathbf{A}$ .
- *Dictionary update*:  $\mathbf{A}$  is kept constant, the energy function is minimized with respect to  $\mathbf{D}$ .

The algorithm alternates between sparse coding and dictionary update until convergence. Different dictionary learning methods use different strategies to perform these two steps, of which the sparse coding step is essentially the problem in Eq. (6). For example, the original method [39] uses convex optimization for sparse coding and gradient descent for dictionary update. A fast online learning algorithm was recently proposed [34] by using a subset of the training data for the optimization problem in Eq. (11) and then augmenting the subset

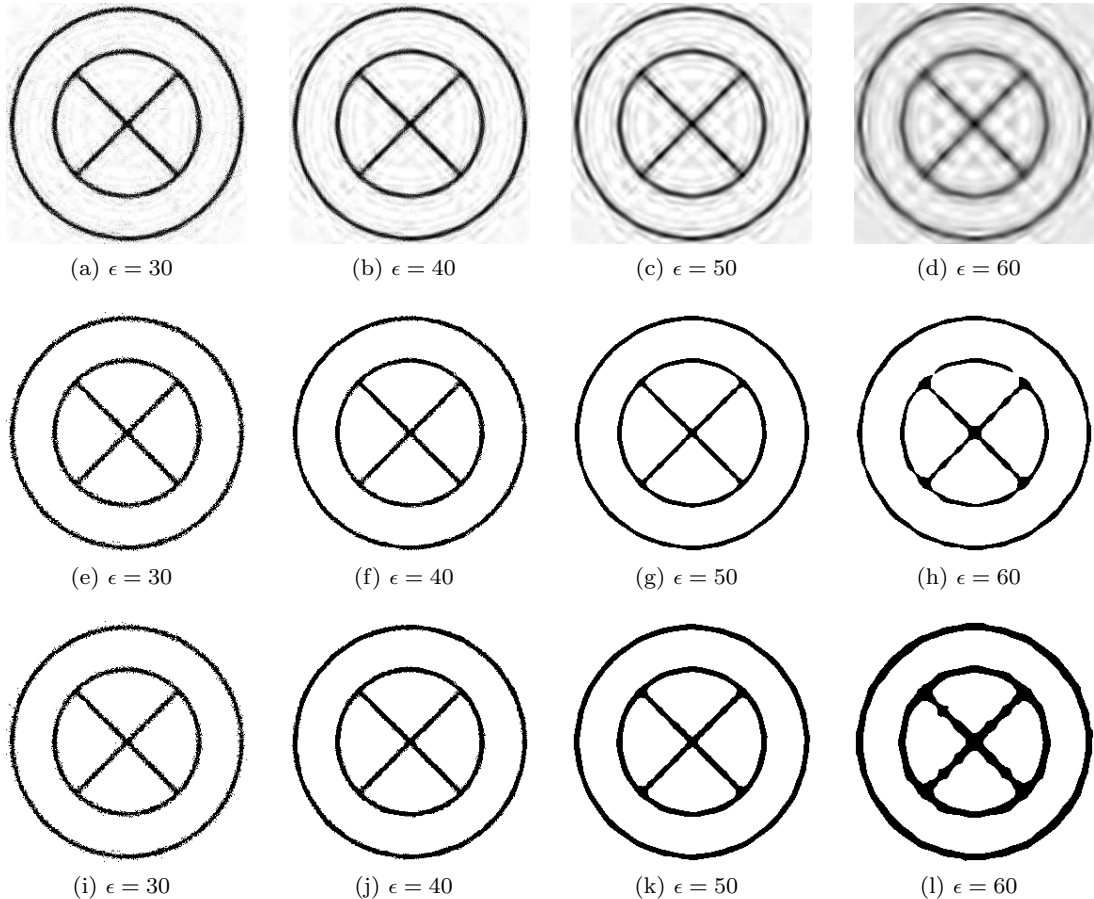


Fig. 8: Influence of the value of  $\epsilon$  on the estimated images using the noisy image in Fig. 5a at  $\epsilon = 30, 40, 50, 60$ . *Top row*: estimated images in grayscale, *middle row*: denoised images in binary using a fixed threshold of 0.5, *bottom row*: denoised images in binary using Otsu's threshold.

with new training samples to compute a new solution using the previous solution as initialization.

Provided that  $\mathbf{D}$  has been learned by solving the problem in Eq. (11), denoising  $\mathbf{X}$  leads to solving

$$\hat{\mathbf{A}} = \underset{\mathbf{A}}{\operatorname{argmin}} \sum_{i=1}^n \lambda \|\boldsymbol{\alpha}^i\|_1 \quad \text{subject to} \quad \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_2 \leq \epsilon,$$

which is in fact an aggregate of Eq. (9) for all images  $\mathbf{x}_i \in \mathbf{X}$ . Accordingly, the denoised images are  $\hat{\mathbf{X}} = \mathbf{D}\hat{\mathbf{A}}$ .

Since learning  $\mathbf{D}$  directly from data is very expensive when  $\mathbf{D}$  has a large size, it is common in practice to learn  $\mathbf{D}$  from a set of some small-sized representative ‘‘patches’’ extracted from the noisy image. When  $\mathbf{D}$  has been learned, a denoised version of each patch from the noisy image is then computed from  $\mathbf{D}$  using Eq. (9). The final denoised image in grayscale is formed by tiling and averaging all the denoised patches according to their extraction pattern from the noisy image. In this work, the patch size is chosen to be  $16 \times 16$  to balance performance and complexity. Moreover, due

to the adoption of patch-based denoising when  $\mathbf{D}$  is learned from data to avoid high computational complexity, the linear relationship conjecture in Subsection 4.4 becomes incorrect. This is because of the locality of patch data, or the locality of dictionary atoms, that  $\mathbf{D}$  cannot represent the noisy image as a whole. Fig. 9 illustrates some atoms of size  $16 \times 16$  in a dictionary learned from some patches extracted from the noisy image in Fig. 5a. It can be observed that the learned dictionary contains atoms that represent oriented edges well in the graphical document images.

For convenience, in the remainder of this paper BPDN with the dictionary defined as curvelets is denoted as BPDN-fixed whereas BPDN with the dictionary learned from data is denoted as BPDN-learned.

#### 4.6 Algorithm summary

A summary of the sparsity-based denoising algorithm is given in Algorithm 1. The sparse coding step to solve

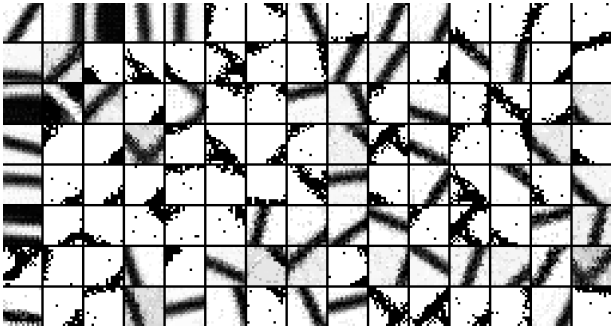


Fig. 9: Some atoms of size  $16 \times 16$  in a dictionary learned from some patches extracted from the image in Fig. 5a.

Eq. (9) is carried out by using the orthogonal matching pursuit [41]. The dictionary learning step is performed by using the K-SVD algorithm [1]. For BPDN-fixed, the learning step is not needed since the dictionary  $\mathbf{D}$  is defined analytically as curvelets. In practice, we use an efficient implementation of K-SVD for dictionary learning and batch-OMP for sparse coding [44]. A thorough analysis of the complexity of sparse coding and dictionary learning steps has also been given in [44].

It should be noted here that the size of the dictionary  $\mathbf{D}$  in Eq. (9) depends on the length  $wh$  of  $\mathbf{x}$ :  $\mathbf{D}$  has  $wh$  rows and at least  $wh$  columns in order to make  $\mathbf{D}$  redundant. When BPDN-fixed is used,  $wh$  is the number of pixels in the noisy image  $f$  and the size of  $\mathbf{D}$  grows quadratically with the image size. Thus, BPDN-fixed rapidly becomes prohibitively expensive when dealing with noisy images of larger size. In contrast, BPDN-learned works with image patches of size  $16 \times 16$  and  $\mathbf{x}$  has a constant length of 256. The size of  $\mathbf{D}$  in BPDN-learned remains almost the same, regardless of the size of original noisy images. Thus, the size of the noisy image  $f$  only slightly affects the complexity of BPDN-learned. In practice, in order to denoise a large-sized image using BPDN-fixed, the noisy image is first divided into sub-images of small size (for example  $256 \times 256$ ). BPDN-fixed is then used to denoise these sub-images and finally the denoised sub-images are merged together to obtain the denoised image.

## 5 Experimental results

In order to demonstrate the effectiveness of bilevel graphical document image denoising using BPDN with the overcomplete dictionary  $\mathbf{D}$  defined as curvelets or learned from data, three types of experiments have been carried out: one for the validation of the linear relationship between the parameter  $\epsilon$  and NS; one to

show the qualitative effectiveness of the methods over a range of degradation levels and graphical symbol shapes, and the third to demonstrate the superiority, both qualitatively and quantitatively, of the proposed method over comparison ones.

Two metrics will be used for quantitative analysis, image recovery and contour smoothness. Image recovery in this work is measured with the normalized cross correlation between the original image and the resulting image. This is calculated using the formula:

$$\frac{\sum_i \sum_j (A_{ij} - \bar{A})(B_{ij} - \bar{B})}{\sqrt{\left(\sum_i \sum_j (A_{ij} - \bar{A})^2\right) \left(\sum_i \sum_j (B_{ij} - \bar{B})^2\right)}}, \quad (12)$$

where  $A$  and  $B$  are 2D input data of size  $m \times n$  and  $\bar{A}$  and  $\bar{B}$  are the mean values of  $A$  and  $B$ . The normalized cross correlation takes on values in the range  $[-1, 1]$ , with the value 1 occurring when the two images  $A$  and  $B$  are the same (i.e., no error, or perfect reconstruction).

While the use of image recovery as an evaluation criteria is straightforward, contour smoothness is adopted to quantitatively evaluate the capability of comparing methods in producing denoised images of good visual quality. Contour smoothness is a measurement based on raggedness defined in [30]. Raggedness is the standard deviation of the distance of the edge pixels from a line fitted to the edge threshold of the line. In its original form it was applied to straight lines. Here we have adapted it to be a moving average of the raggedness measure over an edge segment 15 pixels long.

### 5.1 The relation $\epsilon(\text{NS})$

A dataset of noisy images has been generated from a ground-truth and noise-free image “symbol017” to be used to evaluate the relationship  $\epsilon(\text{NS})$ . “symbol017” is a graphical symbol image of size  $256 \times 256$  taken from the GREC2005 database [48], its noisy version is given in Fig. 12. This image was degraded at 10 values of NS ranging from 0.2 to 2.0 with increments of 0.2. Assuming that the parameter  $\epsilon$  of BPDN-fixed in Eq. (9) takes the value  $\epsilon^*$  that corresponds to the peak in denoising performance in terms of a measure of fidelity between  $\tilde{\mathbf{x}}$  and  $\mathbf{x}_0$ , the relation  $\epsilon^*(\text{NS})$  thus needs to be established experimentally. The measure of fidelity, denoted by  $\gamma_{\tilde{\mathbf{x}}, \mathbf{x}_0}(\epsilon)$ , employed in this work is the normalized cross-correlation defined as  $\gamma_{\tilde{\mathbf{x}}, \mathbf{x}_0}(\epsilon) = \text{corr2}(\tilde{\mathbf{x}}, \mathbf{x}_0)$ , where  $\text{corr2}(A, B)$  as in Eq. (12).

Illustration of the determination of  $\epsilon^*$  by means of  $\gamma_{\tilde{\mathbf{x}}, \mathbf{x}_0}(\epsilon)$  is given in Fig. 10a where the noisy image in Fig. 12p with NS = 2.0 is taken as the input

**Algorithm 1** Sparsity-based image denoising**Input:** noisy image  $f$ , patch size  $P = p^2$ , dictionary size  $P \times L$ , number of training patches  $n$ , number of iterations  $k$ **Learning for a dictionary  $\mathbf{D}$** 

- 1: Extract  $n$  patches  $\mathbf{x}^{i=1, \dots, n}$  of size  $p \times p$  from  $f$
- 2: Create a training set  $\mathbf{X} = [\mathbf{x}^1, \dots, \mathbf{x}^n]$
- 3: Initialize dictionary  $\mathbf{D}$
- 4: **for**  $j = 1 \dots k$  **do**
- 5:     Solve Eq. (9) to find  $\alpha^i$  for each  $\mathbf{x}^i$
- 6:     **for**  $l = 1 \dots L$  **do**
- 7:         Set  $\mathbf{D}_l = 0$  and define  $I = \{i \mid 1 \leq i \leq n, \alpha_j^i \neq 0\}$
- 8:         Update  $\mathbf{D}_l = \mathbf{d}$  by solving  $\{\mathbf{d}, \alpha\} = \operatorname{argmin}_{\mathbf{d}, \alpha} \|\mathbf{X}_I - \mathbf{D}\mathbf{A}_I - \mathbf{d}\alpha^T\|_2^2$  subject to  $\|\mathbf{d}\|_2 = 1$
- 9:     **end for**
- 10: **end for**

**Denoising using  $\mathbf{D}$** 

- 1: Scan  $f$  to get all patches of size  $p \times p$
- 2: Solve Eq. (9) for each obtained patch  $\mathbf{x}^i$  to find  $\alpha^i$
- 3: Compute  $\hat{\mathbf{x}}^i = \mathbf{D}\alpha^i$  as the denoised version of  $\mathbf{x}^i$
- 4: Merge all  $\hat{\mathbf{x}}^i$  to get  $\hat{f}$

**Output:** denoised image  $\hat{f}$ 

image. To compute the value of  $\gamma_{\tilde{\mathbf{x}}, \mathbf{x}_0}(\epsilon)$  at each possible value of  $\epsilon$ , the  $\ell_1$ -optimization problem in Eq. (9) is solved for  $\hat{\alpha}$  and then the bilevel denoised image  $\tilde{\mathbf{x}} = \mathcal{T}(\hat{\mathbf{x}}) = \mathcal{T}(\mathbf{D}\hat{\alpha})$  can be easily obtained. The plot of  $\gamma_{\tilde{\mathbf{x}}, \mathbf{x}_0}(\epsilon)$  in the case of a fixed thresholding of 0.5 has its maximum value of 0.9439 at  $\epsilon^* = 48$ . This means that if the input noisy image has  $\text{NS} = 2.0$ ,  $\epsilon$  in Eq. (9) should take the value 48 in order to have the “best” denoising performance. It should be noted here that, due to the blunt maxima in  $\gamma_{\tilde{\mathbf{x}}, \mathbf{x}_0}(\epsilon)$ , a small deviation of the selected value of  $\epsilon$  from  $\epsilon^*$  has almost no effect on the performance of BPDN-fixed.

Having determined the value of  $\epsilon^*$  for each image of a certain NS, the relation  $\epsilon^*(\text{NS})$  is established. This was repeated three more times, by generating three more sets of 10 degraded images, with different instances of noise. These four relations are plotted separately in Fig. 10b. It can be seen that an image of higher NS requires a larger value of  $\epsilon^*$  for optimal performance. In addition,  $\epsilon^*$  has a nearly linear relationship with NS for all four subsets. A narrow band formed by  $\epsilon^*(\text{NS})$  also means that the standard deviation of  $\epsilon^*$  is reasonably small. Combining this fact with the blunt maxima in  $\gamma_{\tilde{\mathbf{x}}, \mathbf{x}_0}(\epsilon)$ , it thus can be concluded that if the parameter  $\epsilon$  is estimated from the relation  $\epsilon^*(\text{NS})$ , the proposed method with curvelets as the dictionary performs almost at its full potential. Similar results were seen for other symbol images from the GREC2005 dataset.

## 5.2 Illustration of results

The proposed method for denoising bilevel graphical document images using BPDN was evaluated on two

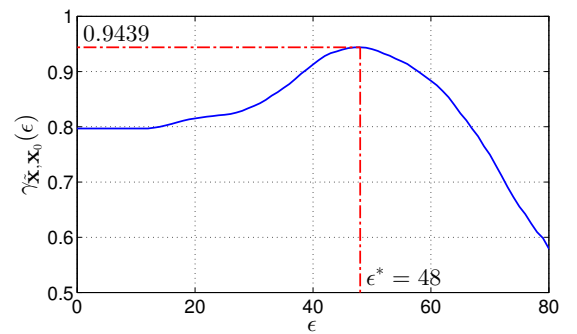
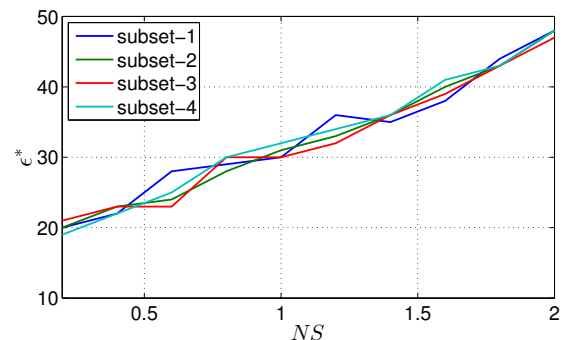
(a) A plot of  $\gamma_{\tilde{\mathbf{x}}, \mathbf{x}_0}(\epsilon)$  at  $\text{NS} = 2.0$ (b)  $\epsilon^*(\text{NS})$  relation

Fig. 10: Determination of the value of the precision parameter  $\epsilon$  in Eq. (9): (a) its optimal value  $\epsilon^*$  is determined by means of image recovery; (b) the linear relationship between  $\epsilon^*$  and NS for the four experimental subsets.

datasets. The first, shown in the previous subsection, was designed to show that results hold over multiple instances of the same levels of noise. The second is designed to illustrate the effectiveness over different

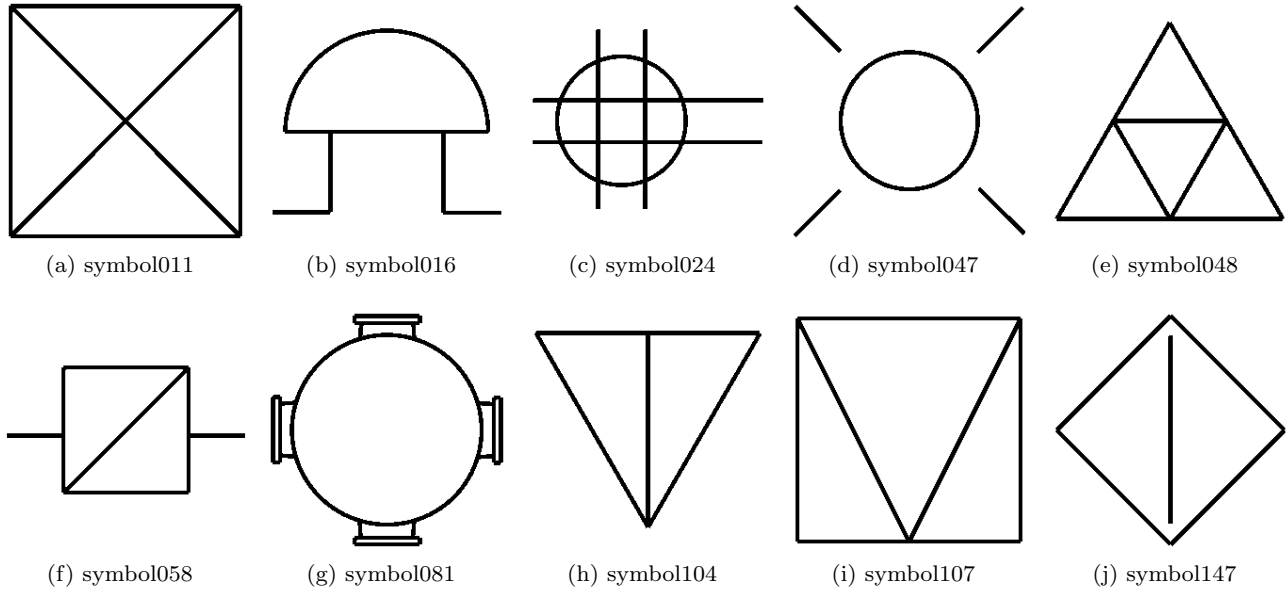


Fig. 11: Some noise-free graphical symbol images from the GREC2005 dataset used to generate noisy images of different NS for a larger evaluation of the proposed method.

shapes of symbols. It is generated from ground-truth and noise-free graphical symbol images 011, 016, 024, 047, 048, 058, 081, 104, 107 and 147, in addition to symbol017 also used in the previous subsection, all from the GREC2005 database. These symbols are shown in Fig. 11. These images are selected due to the existence of all possible graphic contour directions and various configurations of contours that may cause difficulties in denoising.

Figs. 12 and 13 provide examples of denoised graphical symbols by BPDN-fixed and BPDN-learned. Fig. 12 shows symbol017 at four different noise levels  $NS = 0.8, 1.2, 1.6, 2.0$ . Fig 13 shows noisy versions of three different ground-truth images, all at noise levels  $NS=2.0$ . In the case of BPDN-fixed,  $\epsilon = \epsilon^*$  for each case in order to have optimal performance. For BPDN-learned,  $\epsilon$  has a fixed value of 0.75 determined experimentally. In these figures the original noisy images are given in the first column. The corresponding estimated images in grayscale obtained by using BPDN-fixed and BPDN-learned are given in the second and fourth columns, respectively. These images after being converted back to bilevel through thresholding at a 0.5 level are shown in columns three and five. Evidence of directional denoising along noisy contours exists in the estimated images by BPDN-fixed in the second column: edge noise is smoothed out in the direction that is perpendicular to the noisy contour. It is like the images have been filtered locally along the noisy contours by anisotropic filters,

each has its direction coincident with the local direction of the nearest contour. Due to this directional filtering phenomena, the denoised images in binary by BPDN-fixed in the third column are clean and have smooth contours. For BPDN-learned, because of the locality of patch-based denoising, the estimated images by BPDN-learned in the fourth column do not have clear evidence of directional denoising along noisy contours as the estimated images in the second column by BPDN-fixed. However, it can be seen that BPDN-learned does remove noise along the contours and produces grayscale images of very high visual quality. This is because the dictionary used in BPDN is learned directly from the noisy images so that the learned atoms adapt more to the graphical contours. As a result, the denoised images in binary by BPDN-learned in the last column are also clean and have smooth contours.

### 5.3 Comparison with existing methods

To demonstrate the effectiveness of the proposed method based on BPDN, comparison with seven frequently used methods has been carried out:

- Bilevel denoising: median filtering uses a  $3 \times 3$  neighborhood, kFill filtering sets the parameter  $k = 3$ , closing then opening and opening then closing both use a  $3 \times 3$  structuring element.
- Total variation:  $\epsilon_{TV}$  takes the value  $\epsilon_{TV}^*$  that corresponds to optimal performance in terms of normal-

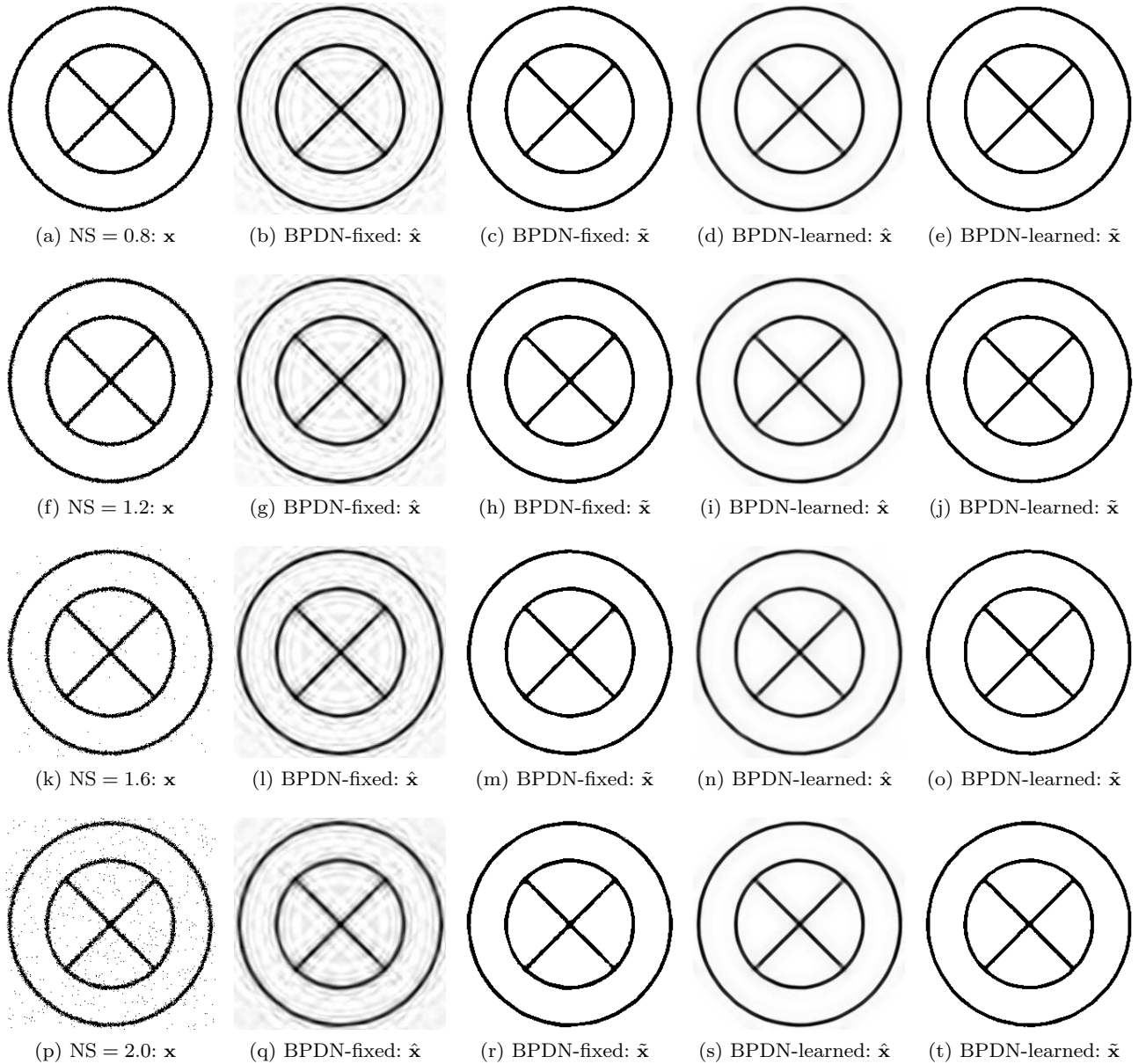


Fig. 12: Some samples of images from the dataset with varying levels of NS and the corresponding denoised images obtained by using BPDN-fixed and BPDN-learned. *First column*: original noisy images, *second column*: estimated images in grayscale using BPDN-fixed, *third column*: denoised images in binary using BPDN-fixed, *fourth column*: estimated images in grayscale using BPDN-learned, *last column*: denoised images in binary using BPDN-learned.

ized cross-correlation. The selection of  $\epsilon_{TV}$  is similar to the selection of  $\epsilon$  in the proposed method.

- Shrinkage: hard-thresholding of curvelet coefficients with one threshold value  $\lambda_{jl}$  is used for all curvelets of scale  $j$  and angle  $l$ .  $\lambda_{jl}$  is computed by applying a forward curvelet transform on an image containing a delta function at its center.
- Diffusion: iterative applications of anisotropic diffusion and coherence enhancing diffusion in sequence with parameters determined by experience.

The proposed method using BPDN with fixed and learned dictionaries and all comparison methods are applied to each degraded image and then normalized cross-correlation and raggedness (described at the start of Section 5) are measured for each resulting denoised image. Samples of denoised images from comparison methods using an image of NS = 2.0 are given in Fig. 14. It can be seen that the images resulting from kFill filtering, median filtering, morphology-based methods, and total variation have ragged edges and bad visual quality

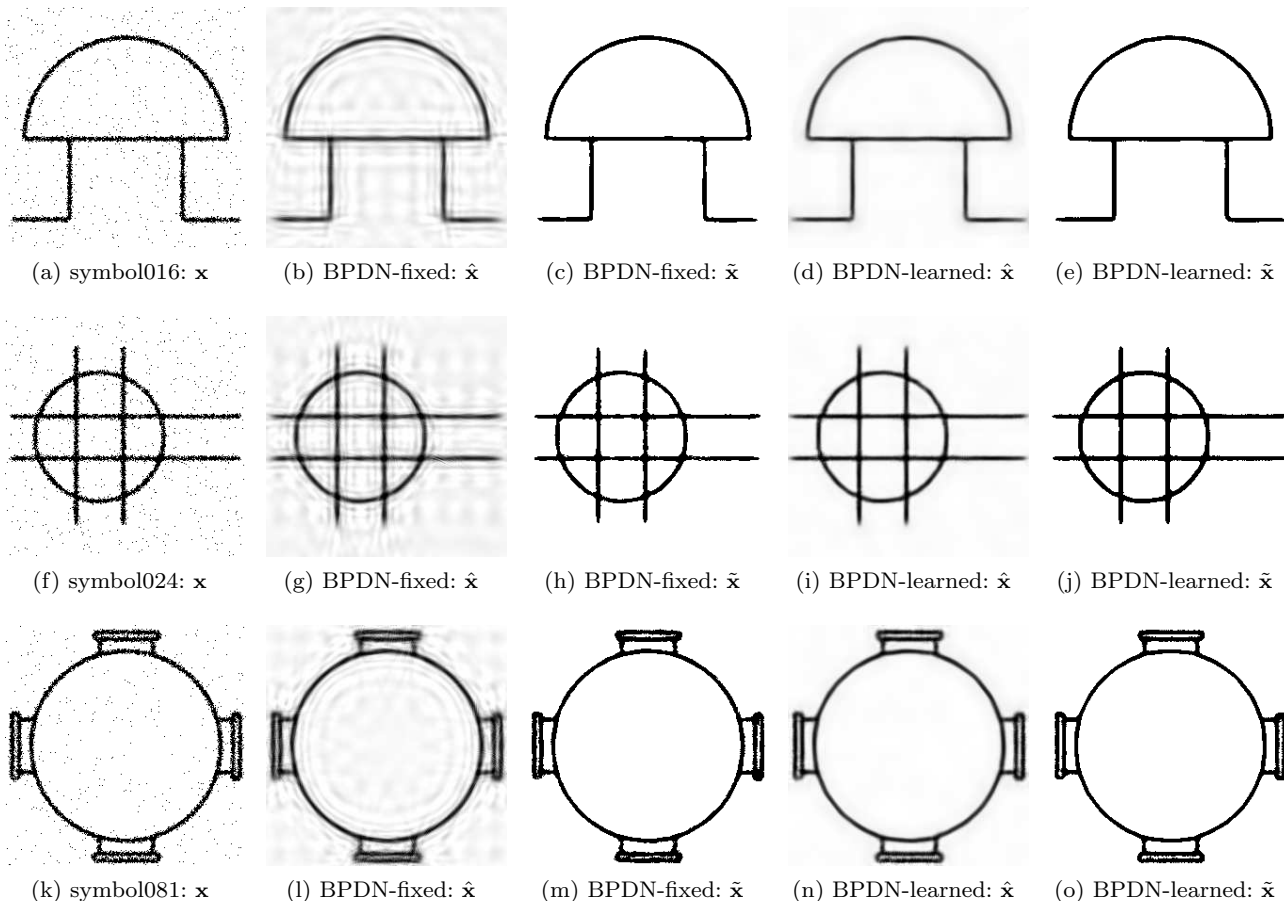


Fig. 13: Samples of noisy images showing effectiveness over a range of shapes. Images at  $NS = 2.0$  and the corresponding denoised images obtained by using BPDN-fixed and BPDN-learned. *First column*: original noisy images, *second column*: estimated images in grayscale using BPDN-fixed, *third column*: denoised images in binary using BPDN-fixed, *fourth column*: estimated images in grayscale using BPDN-learned, *last column*: denoised images in binary using BPDN-learned.

with the worst images resulting from morphology-based methods. Shrinkage, diffusion, and BPDN-based methods produce denoised images of good visual quality. However, the diffusion method has more difficulty in restoring the sharp corners of the contours; shrinkage and BPDN-fixed methods produce graphical strokes of non-uniform width.

For a quantitative comparison, the performance of each method per noise level is computed as the average performance over noisy images of the same noise level. At each noise level, the first dataset has 4 noisy images of symbol017 and the second dataset has 10 noise images of each of the graphical symbols 011, 016, 017, 024, 047, 048, 058, 081, 104, 107 and 147. Thus, the averaging in the second dataset is performed over 110 noisy images. The comparison results are shown in Fig. 15 for the two restoration criteria over a range of noise levels. The left column shows the average over

multiple noise instances of the same shape (symbol 017) and the right column shows the average over a range of different shapes. It is observed that as the noise level ( $NS$ ) increases, the ability to recover the original images decreases and the contour raggedness of the denoised images increases for all methods. The performance of kFill filtering, morphology-based methods, and total variation are similarly bad with open-closing breaking down when the noise level is reasonably high ( $NS > 1$ ). Median filtering has its performance in the middle. Top performance belongs to shrinkage, diffusion, and BPDN-based methods. The decrease in the image recovery and increase in the contour raggedness of these four best methods are nearly constant and are the smallest among all methods for both datasets. Nevertheless, diffusion has slightly worse performance than that of shrinkage and BPDN. BPDN-fixed outperforms shrinkage when the noise level is small to moderate ( $NS < 1.5$ ) and their



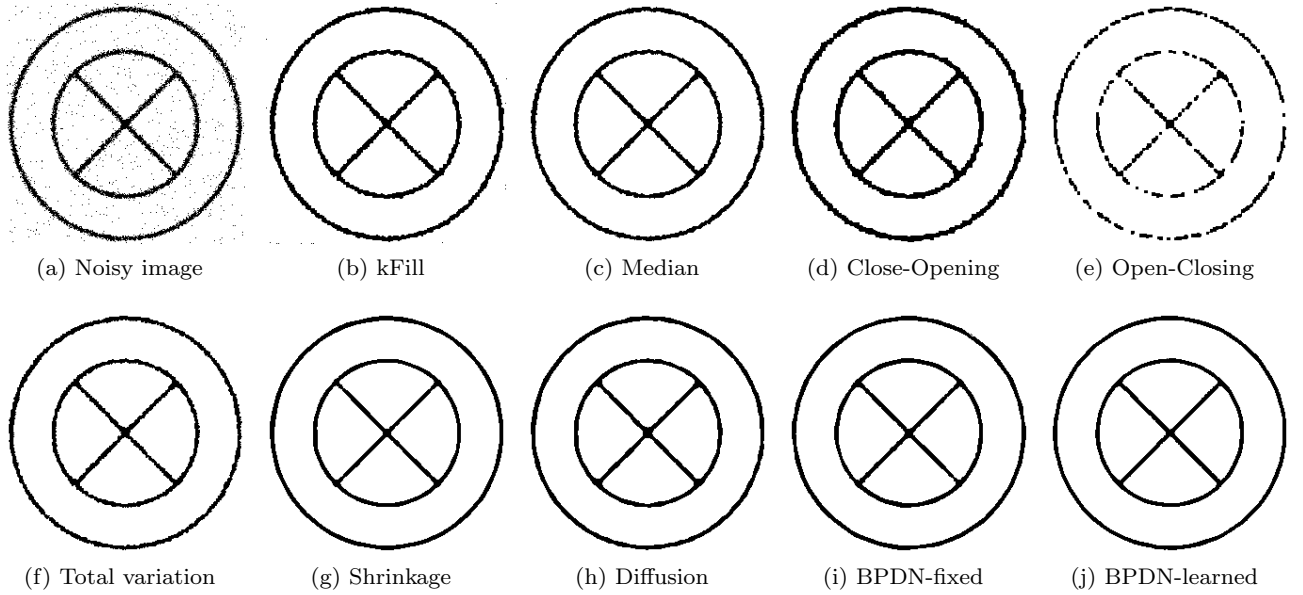


Fig. 14: Samples of denoised images from comparison methods using an image of  $NS = 2.0$ . Shrinkage, diffusion, and sparsity-based methods produce images of good visual quality, whereas the other methods result in images having ragged edges.

performance are comparable when the noise level is high ( $NS \geq 1.5$ ). BPDN-learned results in a denoised image of best recovery with smoothest contours at all values of  $NS$ . It should also be noted from the comparison results that for noisier images, in terms of image recovery and contour raggedness BPDN-learned produces the most significant improvements.

The superiority of BPDN-learned over BPDN-fixed can be explained by the locality of dictionary atoms. The dictionary of BPDN-fixed is constructed from curvelets and each curvelet has a fixed set of location, orientation and scale due to its analytical definition. On the contrary, the dictionary of BPDN-learned is learned directly from the data and thus its atoms will have location, orientation and scale so as to “best” represent the data. The superiority of BPDN-learned over BPDN-fixed and other comparison methods provides clear evidence for the need to carefully design the dictionary in a sparsity-based denoising framework. Fig. 16 shows the 95% confidence interval for the averaged performance of the proposed methods and the two best comparison methods (shrinkage and diffusion) using the aforementioned dataset. The figure shows that the confidence intervals of BPDN-learned are well separated from that of other methods, meaning that BPDN-learned stably outperforms other methods. These results consolidate our conclusion on the superiority of BPDN-learned for graphical document images.

In addition to the results shown in Figs. 12–16 additional experiments were run on symbols of different sizes:  $128 \times 128$ ,  $256 \times 256$ , and  $512 \times 512$  (Fig. 17). Five degradation levels were utilized that implemented  $NS$  levels of 0.2, 1 and 2. The BPDN-fixed and BPDN-learned methods were applied as well as the comparison methods. Image recovery and contour smoothness were measured for each case. For all the degradation levels at all the resolutions the sparsity-based methods continued to exhibit lower raggedness and higher image recovery than the other comparison methods, with results very similar to those in Figs. 12–15. The performance of each denoising method improves as the resolution increases. At each noise level and for each denoising method, when the image size increases the image recovery value increases and the contour raggedness value decreases. This phenomena can be explained by the aliasing effect that exists when the graphical contours are discretized. At higher resolution, this effect becomes less serious and denoising methods have less difficulty in recovering the original contours. However and more importantly, the relative performance of denoising methods on this dataset is similar to that on previous datasets. Sparsity-based denoising methods still provide the best performance. This demonstrates the robustness of sparsity-based denoising methods to resolution.

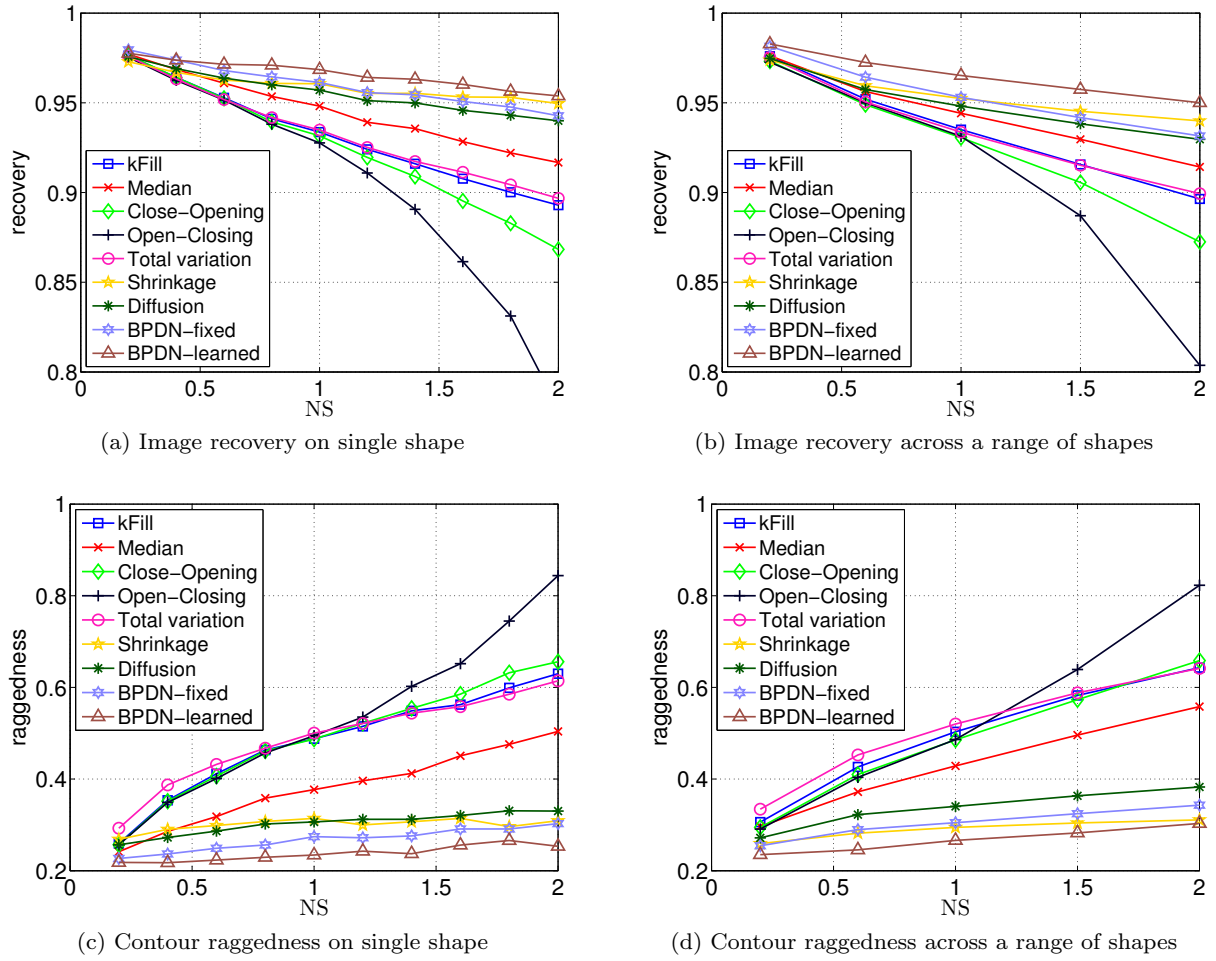


Fig. 15: Performance evaluation of the proposed and the seven comparison denoising methods in terms of image recovery (top row) and contour raggedness (bottom). The performance is compared on an individual shape (left column) and across a range of shapes (right column).

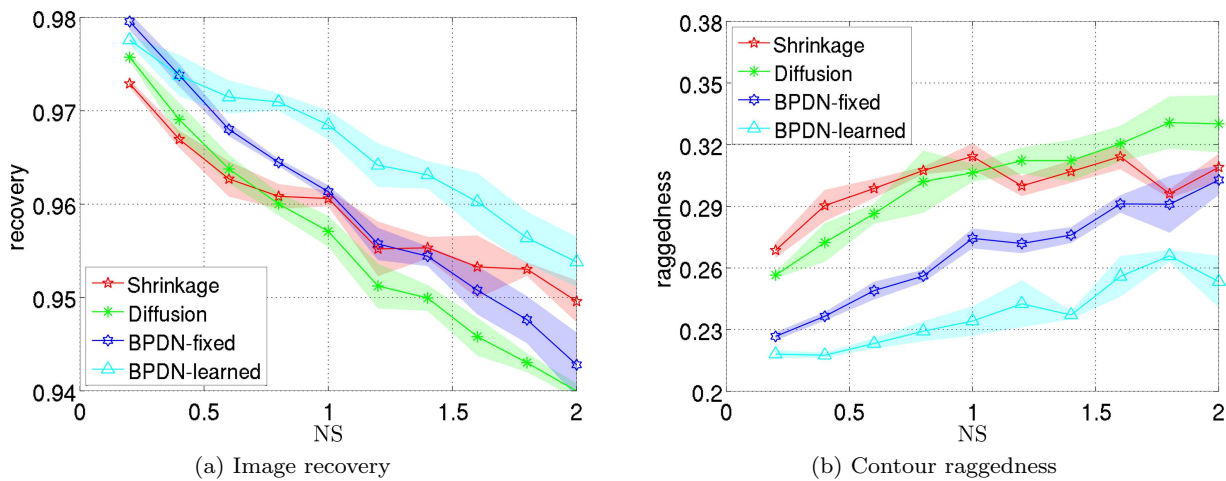


Fig. 16: The 95% confidence interval for the averaged performance in terms of image recovery (left) and contour raggedness (right) of the proposed methods and the two best comparison methods (shrinkage and diffusion) using the experimental dataset SetA.

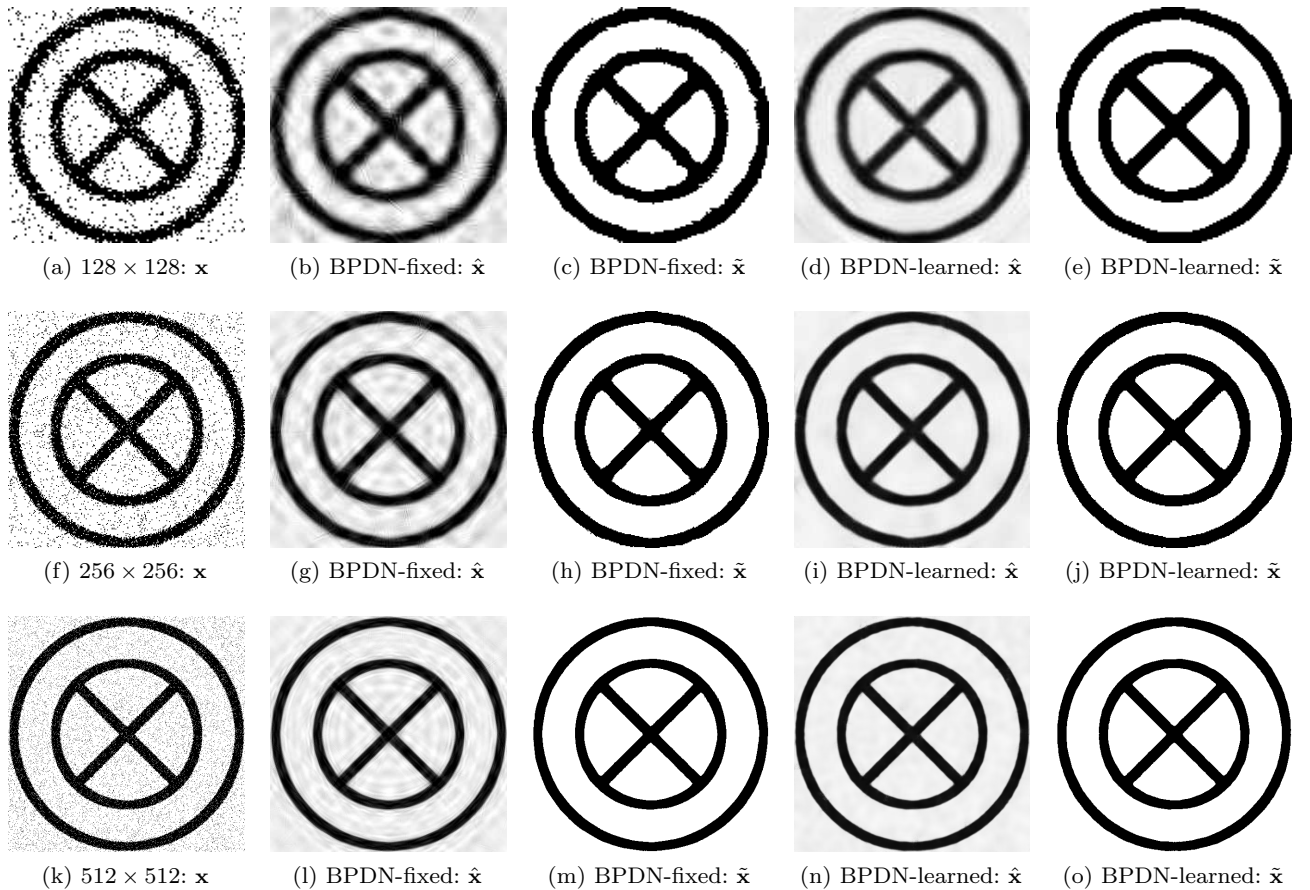


Fig. 17: Some samples of images from the multiresolution dataset at  $NS = 2$  and the corresponding denoised images obtained by using BPDN-fixed and BPDN-learned. *First row*:  $128 \times 128$  images, *second row*:  $256 \times 256$  images, *third row*:  $512 \times 512$  images. *First column*: original noisy images, *second column*: estimated images in grayscale using BPDN-fixed, *third column*: denoised images in binary using BPDN-fixed, *fourth column*: estimated images in grayscale using BPDN-learned, *last column*: denoised images in binary using BPDN-learned.

Performance of sparsity-based denoising algorithms has also been evaluated on some degraded images of engineering drawings of size  $1950 \times 2700$  and example results are given in Fig. 18. Shrinkage, diffusion, and sparsity-based methods still produce images of good visual quality, whereas the other methods result in images having ragged edges. Although shrinkage produces denoised images that have smooth contours, it fails in recovering dashed lines and contour intersection points. Dashed lines become continuous and some intersecting lines get disconnected after denoising. In addition, the graphics and text components recovered by shrinkage usually have distorted shapes. BPDN-fixed produces denoised images that are somewhat similar to the ones produced by shrinkage. This can be explained by the same dictionary used by both BPDN-fixed and shrinkage. Diffusion is better than shrinkage when dealing with dashed lines and intersection points. However,

when compared with BPDN-learned, the visual quality of denoised image obtained by BPDN-learned is better than that obtained by diffusion. These experimental results demonstrate clearly the performance and usefulness of BPDN-learned for large-sized scanned graphical document images.

## 6 Conclusion

A new parametric method for edge noise removal from graphical document images has been presented in this paper using sparse representation by means of basis pursuit denoising. When the overcomplete dictionary is defined based on curvelets, Noise Spread can be used as the input parameter to select the precision parameter  $\epsilon$  of the method due to the linear relationship between  $\epsilon$  and  $NS$ . In the case of the learned dictionary, due to the locality of small-sized patch data, this linear

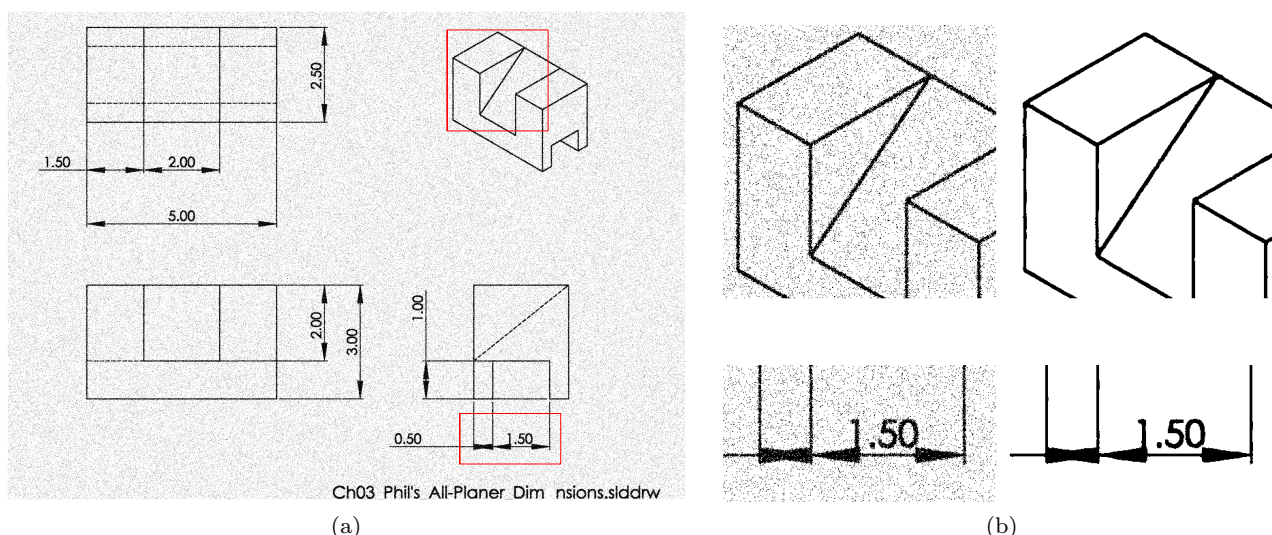


Fig. 18: A  $1950 \times 2700$  degraded image of an engineering drawing (a) with sample noisy and restored images obtained by using BPDN-learned (b).

relationship does not hold anymore. However, it is anticipated that if the patch size is equal to the image size, or in other words  $\mathbf{D}$  is learned directly from the whole image data, the relation  $\epsilon^*(\text{NS})$  becomes linear again. This is because  $\epsilon$  in that case is simply the upper bound of the Hamming distance between the binary denoised image and its corresponding noisy one. The superiority of the proposed method over comparison ones in terms of image recovery and contour raggedness has been validated by experimental results.

It is possible that other image restoration methods, such as those discussed in this paper and others, might also benefit from measurement of NS and include it in their parameter setting. Investigating this is planned as future work.

## References

- Aharon, M., Elad, M., Bruckstein, A.: K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing* **54**(11), 4311–4322 (2006)
- Arce, G.R., Bacca, J., Paredes, J.L.: Nonlinear filtering for image analysis and enhancement. In: A.C. Bovik (ed.) *The Essential Guide to Image Processing*, chap. 12, pp. 263–291. Elsevier (2009)
- Baird, H.S.: Document image defect models. In: H.S. Baird, K. Yamamoto, H. Bunke (eds.) *Structured Document Image Analysis*, pp. 546–556. Springer-Verlag (1992)
- Barney Smith, E.H.: Modeling image degradations for improving OCR. In: *Proceedings of the 16th European Signal Processing Conference* (2008)
- Barney Smith, E.H., Qiu, X.: Statistical image differences, degradation features, and character distance metrics. *International Journal on Document Analysis and Recognition* **6**(3), 146–153 (2003)
- Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences* **2**(1), 183–202 (2009)
- Boncellet, C.: Image noise models. In: A.C. Bovik (ed.) *The Essential Guide to Image Processing*, chap. 7, pp. 143–167. Elsevier (2009)
- Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation* **4**(2), 490–530 (2005)
- Candès, E.J., Donoho, D.L.: New tight frames of curvelets and optimal representations of objects with piecewise  $C^2$  singularities. *Communications on Pure and Applied Mathematics* **57**(2), 219–266 (2002)
- Candès, E.J., Guo, F.: New multiscale transforms, minimum total variation synthesis: applications to edge-preserving image reconstruction. *Signal Processing* **82**(11), 1519–1543 (2002)
- Chandrasekaran, V., Wakin, M.B., Baron, D., Baraniuk, R.G.: Representation and compression of multidimensional piecewise functions using surflets. *IEEE Transactions on Information Theory* **55**(1), 374–400 (2009)
- Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing* **20**(1), 33–61 (1998)
- Daubechies, I., DeVore, R., Fornasier, M., Güntürk, C.S.: Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics* **63**(1), 1–38 (2010)
- Davis, G., Mallat, S., Avellaneda, M.: Adaptive greedy approximations. *Constructive Approximation* **13**(1), 57–98 (1997)
- Demanet, L., Ying, L.: Wave atoms and sparsity of oscillatory patterns. *Applied and Computational Harmonic Analysis* **23**(3), 368–387 (2007)
- Do, M.N., Vetterli, M.: The finite ridgelet transform for image representation. *IEEE Transactions on Image Processing* **12**(1), 16–28 (2003)

17. Do, M.N., Vetterli, M.: The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions on Image Processing* **14**(12), 2091–2106 (2005)
18. Donoho, D.L.: For most large underdetermined systems of linear equations the minimal  $\ell_1$ -norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics* **59**(7), 797–829 (2006)
19. Donoho, D.L., Huo, X.: Beamlets and multiscale image analysis. In: T.J. Barth, T. Chan, R. Haimes (eds.) *Multiscale and Multiresolution Methods*, pp. 149–196. Springer (2001)
20. Donoho, D.L., Johnstone, I.M.: Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association* **90**(432), 1200–1224 (1995)
21. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. *Annals of Statistics* **32**(2), 407–499 (2004)
22. Elad, M.: *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer (2010)
23. Elad, M., Matalon, B., Zibulevsky, M.: Image denoising with shrinkage and redundant representations. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1924–1931 (2006)
24. Freeman, W.T., Adelson, E.H.: The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(9), 891–906 (1991)
25. Guo, K., Labate, D.: Optimally sparse multidimensional representation using shearlets. *SIAM Journal on Mathematical Analysis* **39**(1), 298–318 (2007)
26. Hamming, R.W.: Error detecting and error correcting codes. *Bell System Technical Journal* **29**(2), 147–160 (1950)
27. Hoang, T.V.: *Image representations for pattern recognition*. Ph.D. thesis, University of Nancy (2011)
28. Hoang, T.V., Barney Smith, E.H., Tabbone, S.: Edge noise removal in bilevel graphical document images using sparse representation. In: *Proceedings of the 18th IEEE International Conference on Image Processing*, pp. 3610–3613 (2011)
29. Hoang, T.V., Tabbone, S.: Text extraction from graphical document images using sparse representation. In: *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pp. 143–150 (2010)
30. ISO/IEC 13660:2001: Information technology - Office equipment - Measurement of image quality attributes for hardcopy output - Binary monochrome text and graphic images. ISO, Geneva, Switzerland (2001)
31. Jansen, M.H.: *Noise Reduction by Wavelet Thresholding*. Springer (2001)
32. Lee, T.S.: Image representation using 2D Gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(10), 959–971 (1996)
33. Ma, J., Plonka, G.: The curvelet transform. *IEEE Signal Processing Magazine* **27**(2), 118–133 (2010)
34. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research* **11**, 19–60 (2010)
35. Maragos, P.: Morphological filtering. In: A.C. Bovik (ed.) *The Essential Guide to Image Processing*, chap. 13, pp. 293–321. Elsevier (2009)
36. McGillivray, C., Hale, C., Barney Smith, E.H.: Edge noise in document images. In: *Proceedings of the 3rd Workshop on Analytics for Noisy Unstructured Text Data*, pp. 17–24 (2009)
37. Meyer, F.G., Coifman, R.R.: Brushlets: a tool for directional image analysis and image compression. *Applied and Computational Harmonic Analysis* **4**(2), 147–187 (1997)
38. O’Gorman, L.: Image and document processing techniques for the RightPages electronic library system. In: *Proceedings of the 11th International Conference on Pattern Recognition*, vol. 2, pp. 260–263 (1992)
39. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381**, 607–609 (1996)
40. Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics* **9**(1), 62–66 (1979)
41. Pati, Y.C., Rezaifar, R., Rezaifar, Y.C.P.R., Krishnaprasad, P.S.: Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In: *Proceedings of the 27th Annual Asilomar Conference on Signals, Systems, and Computers*, pp. 40–44 (1993)
42. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**(7), 629–639 (1990)
43. Rajashekar, U., Simoncellis, E.P.: Multiscale denoising of photographic images. In: A.C. Bovik (ed.) *The Essential Guide to Image Processing*, chap. 11, pp. 241–261. Elsevier (2009)
44. Rubinstein, R., Zibulevsky, M., Elad, M.: Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. *Tech. Rep. CS-2008-08, Technion* (2008)
45. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* **60**, 259–268 (1992)
46. Starck, J.L., Candès, E.J., Donoho, D.L.: The curvelet transform for image denoising. *IEEE Transactions on Image Processing* **11**(6), 670–684 (2002)
47. Strong, D., Chan, T.: Edge-preserving and scale-dependent properties of total variation regularization. *Inverse Problems* **19**(6), S165–S187 (2003)
48. Valveny, E., Dosch, P., Fornés, A., Escalera, S.: Report on the third contest on symbol recognition. In: *Proceedings of the 7th International Workshop on Graphics Recognition*, pp. 321–328 (2007)
49. Weaver, J.B., Xu, Y., Healy, D.M., Cromwell, L.D.: Filtering noise from images with wavelet transforms. *Magnetic Resonance in Medicine* **21**(2), 288–295 (1991)
50. Weickert, J.: Coherence-enhancing diffusion filtering. *International Journal of Computer Vision* **31**(2–3), 111–127 (1999)
51. Wipf, D., Rao, B.: Sparse Bayesian learning for basis selection. *IEEE Transactions on Signal Processing* **52**(8), 2153–2164 (2004)
52. Yu, D., Yan, H.: An efficient algorithm for smoothing, linearization and detection of structural feature points of binary image contours. *Pattern Recognition* **30**(1), 57–69 (1997)