International Scholarly Research Network
ISRN Signal Processing
Volume 2011, Article ID 672353, 9 pages
doi:10.5402/2011/672353

Research Article

Edge-Detection in Noisy Images Using Independent Component Analysis

Kaustubha Mendhurwar,[1] Shivaji Patil,[1] Harsh Sundani,[2] Priyanka Aggarwal,[2] and Vijay Devabhaktuni[2]

[1] *Department of ECE, Concordia University, 1455 de Maisonneuve West, Montreal, QC, Canada H3G 1M8*
[2] *EECS Department, University of Toledo, MS 308, 2801 W. Bancroft Street, Toledo, OH 43606, USA*

Correspondence should be addressed to Priyanka Aggarwal, aggarwalpp@gmail.com

Received 20 January 2011; Accepted 21 February 2011

Academic Editor: F. Palmieri

Edges in a digital image provide important information about the objects contained within the image since they constitute boundaries between objects in the image. This paper proposes a new approach based on independent component analysis (ICA) for edge-detection in noisy images. The proposed approach works in two phases—the training phase and the edge-detection phase. The training phase is carried out only once to determine parameters for the ICA. Once calculated, these ICA parameters can be employed for edge-detection in any number of noisy images. The edge-detection phase deals with transitioning in and out of ICA domain and recovering the original image from a noisy image. Both gray scale as well as colored images corrupted with Gaussian noise are studied using the proposed approach, and remarkably improved results, compared to the existing edge-detection techniques, are achieved. Performance evaluation of the proposed approach using both subjective as well as objective methods is presented.

1. Introduction

In typical images, edges characterize the object boundaries with sharp changes in intensity levels. Edges are useful for segmentation, registration, and identification of objects in a given scene. Most of these operations have edge-detection as the preprocessing step. Consequently, the success of these image processing tasks depends strictly on the performance of edge-detection step. Detection of these edges, therefore, not only helps in more accurate representation of an image, but it also significantly simplifies its processing. The edge detection usually involves calculation of derivative of the image intensity function at a given pixel location owing to the fact that image intensity shows sudden changes at edges. Pixels with relatively higher magnitude of derivative of the image intensity function are classified into edge pixels. To this end, Gradient and Laplacian operators/masks such as Prewitt, Roberts, Sobel, Canny [1–3] are usually employed for the purpose of edge-detection. These operators work well for specific cases; however, they fail for others. For instance, the Prewitt edge detector works quite well for digital images corrupted with Poisson noise, whereas its performance decreases sharply for other kinds of noise [4]. Moreover, these operators/masks are chosen independent of the image under consideration. As such, the performance of these operators masks degrades considerably with the increase in noise levels in images [2]. Unfortunately, digital images are inevitably degraded by noise during acquisition and/or transmission. Furthermore, most of the edge-detection operators assume that images contain regions separated by clear boundaries. However, this assumption is not valid for images corrupted with noise. Alternatively, better results can be obtained if the images are denoised employing an appropriate filter (e.g., linear/smoothing filter, nonlinear filter) before applying the operators/masks [2, 4]. However, filters while smoothing out the noise also remove genuine high-frequency edge features, degrade localization and detection of low-contrast edges, and, therefore are incapable of providing satisfactory results [2–4].

This paper proposes a new approach for edge-detection in noisy images. Proposed approach employs the independent component analysis (ICA) concept [5–7] in conjunction with the traditional edge-detection methods to obtain much better results for images corrupted with Gaussian noise. ICA has been previously applied for the purpose of edge-detection [8–11]. In the proposed approach, basic edge-detection method (i.e., Canny) is used along with ICA for the first time to improve the accuracy of edge-detection in noisy gray-scale and colored images. ICA provides a linear representation of non-Gaussian (i.e., super-Gaussian and sub-Gaussian) data making all the components statistically independent [12–16]. The proposed approach functions in two phases. In the first phase, FastICA algorithm which is a neural network learning algorithm based on a Gradient approach is used to find the separation matrix. In second phase, this obtained separation matrix is used for transitioning in and out of ICA domain, thereby facilitating denoising of noisy images. We apply the proposed approach to several gray-scale and colored images corrupted with high levels of Gaussian noise. Edge-detection results obtained from the proposed approach illustrate better performance than those obtained using Canny along with the nonlocal means filter.

A brief introduction of ICA followed by the information regarding basis functions, fast ICA algorithm, and edge-detection are given in Section 2. Section 3 presents the proposed approach and discusses key points involved in obtaining the improved results. Illustration of the proposed approach applied to gray scale as well as colored images is presented in Section 4, and conclusions with possible future work are included in Section 5.

2. Independent Component Analysis (ICA)

Typically, ICA is used for blind source separation [14], which facilitates separation of a set of signals from mixed signals, with very little information about the source signals or the mixing process. In contrast to the correlation-based transformations, like principal component analysis (PCA) [15], ICA decorrelates the signals (2nd-order statistics) and also reduces higher-order statistical dependencies, in attempt to make signals as independent as possible [7]. Since all the natural images contain similar statistical information, a set of noise-free images can be used for the training phase of ICA [12]. The transform obtained from ICA is then employed for denoising a noisy image.

2.1. Definition of ICA. Let $x_{m\times 1} = [x_1, x_2, x_3, \ldots, x_m]$ be a linear mixture vector with m linear mixtures of n independent sources, $s_{n\times 1} = [s_1, s_2, s_3, \ldots, s_n]$. The relation between mixture vector $\mathbf{x}$ and the source vector $\mathbf{s}$ can be expressed as

$$x_j = a_{j1}s_1 + a_{j2}s_2 + a_{j3}s_3 + \cdots + a_{jn}s_n, \quad \text{for } j = 1, 2, \ldots, m, \tag{1}$$

where $\mathbf{A}$ is called the mixing matrix of size $(m \times n)$, and each column $\mathbf{a}_j$ is called basis function. As such, basis functions project the independent sources to produce linear mixtures. Thus, (1) can be expressed mathematically as

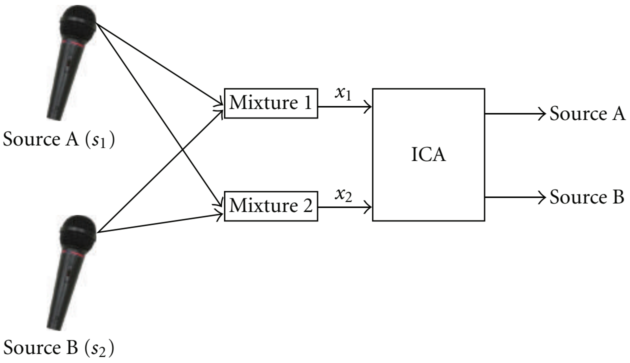


Figure 1: Representative block diagram of ICA for the case of two sources.

$$\mathbf{x} = \mathbf{A}\mathbf{s}. \tag{2}$$

The above model (2) is known as the ICA model. Figure 1 illustrates a block diagram for ICA where $\mathbf{s}$ has two elements – s_1 and s_2. Mixtures x_1 and x_2 obtained from $\mathbf{s}$ are fed to ICA as inputs. Therefore, only mixture vector $\mathbf{x}$ is available, from which $\mathbf{A}$ and $\mathbf{s}$ are estimated by the ICA.

ICA starts with an assumption that elements of vector $\mathbf{s}$ are statistically independent. The distributions of elements of $\mathbf{s}$, although unknown, are assumed to be non-Gaussian. Furthermore, for the sake of simplicity, mixing matrix $\mathbf{A}$ is assumed to be a square matrix. The objective of ICA is to find a matrix $\mathbf{W}$ which, when multiplied with the observed mixture vector $\mathbf{x}$, gives the source vector $\mathbf{s}$, that is,

$$\mathbf{s} = \mathbf{W}\mathbf{x}. \tag{3}$$

$\mathbf{W}$ is referred to as the separation matrix, and its inverse results in the mixing matrix $\mathbf{A}$. There are numerous ways of finding matrix $\mathbf{W}$, but we employed "fixed-point FastICA algorithm" as it is computationally efficient and robust [5–7].

2.2. Implementation of Fixed-Point FastICA. Sample image data obtained from the noise-free images is whitened, in order to make $\mathbf{W}$ and $\mathbf{A}$ unitary and then fed to FastICA algorithm. The computation steps for matrix $\mathbf{W}$ are detailed below.

Step 1. Random image samples of size 8 × 8 each, from the database that we have created, are fed to FastICA.

Step 2. Matrix $\mathbf{W}$ (size 64×64) is obtained from FastICA that performs orthogonalization during each iteration.

Step 3. Each row vector of $\mathbf{W}$ corresponds to a basis function, and kurtosis k is computed, as explained in the next section, for all the basis functions.

Step 4. Row vectors of $\mathbf{W}$, that is, the basis functions, are then arranged in descending order of computed k value.

In this paper, we use $m = n = 64$. As such, both $\mathbf{W}$ and $\mathbf{A}$ have size of 64×64. Obtained matrix $\mathbf{W}$ from FastICA is an orthogonal matrix, and therefore $\mathbf{A}$ is simply $\mathbf{W}^T$. A detailed explanation of the use of FastICA algorithm and MATLAB code can be obtained from [17].

2.3. Edge-Detection. Signal mixtures tend to have Gaussian probability density functions (PDFs), while source signals have non-Gaussian pdf [18]. Hence, ICA is optimized to obtain non-Gaussian signals at the output. In practice, there are two types of non-Gaussian signals, commonly referred to as Platykurtotic (i.e., super-Gaussian) and Leptokurtotic (i.e., sub-Gaussian) [18]. The super-Gaussian signals have most of the values centered at zero, unlike sub-Gaussian signals. This implies that the former signals have relatively higher peaks compared to the latter. Peaky nature of the signals is formally described by what is called kurtosis of the signal's PDF, which is the 4th moment of PDF. The kurtosis of row vector $\mathbf{w}$ of matrix $\mathbf{W}$ is computed as [18]

$$k = \frac{(1/n)\sum_{i=1}^{n}\left(w_i - w_m\right)^4}{(1/n)\sum_{i=1}^{n}\left(w_i^2 - w_m^2\right)^2} - 3, \qquad (4)$$

where w_m is the mean of $\mathbf{w}$. Constant "3" guarantees that the Gaussian, super-Gaussian (sparse), and sub-Gaussian signals have, respectively, zero, positive, and negative values of k. Essentially, kurtosis provides a measure for the 4th central moment, which takes a given signal's variance into account. The proposed algorithm (as explained below) arranges the rows of $\mathbf{W}$ in descending order of their k values. This is to select the sparse components of an image as they represent edges of the image.

3. Proposed Approach

As mentioned earlier, the proposed approach works in two phases—the training phase and the edge-detection phase. Training phase is carried out only once to find the matrix $\mathbf{W}$, which is then employed for edge-detection in any number of noisy images. Both the phases are described below.

3.1. Training Phase. Training phase of the proposed approach consists of two main steps, that is, database creation and ICA followed by kurtosis computation.

3.1.1. Database Creation. Selective images are taken from a website [19] to serve as a database. As can be seen in Figure 2, natural images are selected, since the ICA filters have more sparsely distributed (kurtotic) outputs on natural scenes [20].

3.1.2. ICA and Kurtosis Computation. A detailed flow chart of the training phase is given in Figure 3(a). Random image samples, each of size 8×8, are selected and fed to FastICA as input. FastICA algorithm, as explained earlier, computes the separation matrix $\mathbf{W}$ of size 64×64. This $\mathbf{W}$ is orthogonalized during each iteration to avail several benefits stated in [20]. Reduced arithmetic operations, and relatively easier matrix inversion, are a few of the benefits. Orthogonalization of $\mathbf{W}$ is carried out using

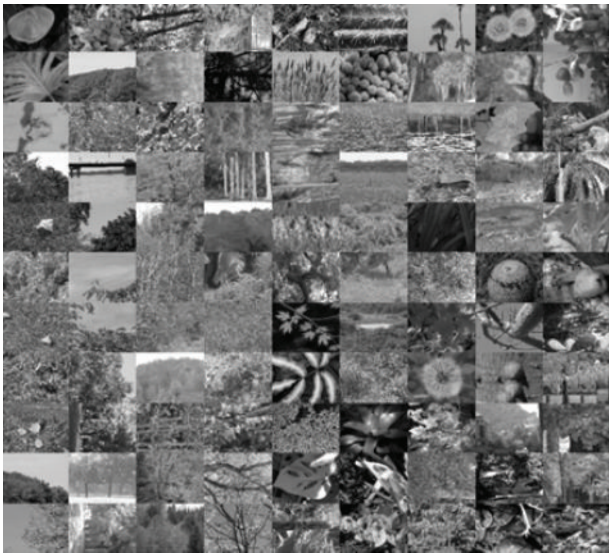


Figure 2: Images selected for the training phase [19].

$$\mathbf{W} = \mathrm{real}\left(\left(\mathbf{W}^T\mathbf{W}\right)^{-1/2}\right)\mathbf{W}. \qquad (5)$$

Kurtosis k (i.e., the degree to which a statistical frequency curve is peaked) is then computed for each row vector $\mathbf{w}$ of matrix $\mathbf{W}$. Positive values of k are required for sparsity of the elements. Constant "3" in (4) ensures high sparseness of the elements. The row vectors of $\mathbf{W}$ are arranged in descending order of their corresponding computed k values. This ensures use of sparse components that represent edges. Separation matrix $\mathbf{W}$ obtained after above-mentioned processing can be used for edge-detection in any noisy image [20].

3.2. Edge-Detection Phase. The flowchart for edge-detection in noisy images is given in Figure 3(b). This phase makes use of the separation matrix $\mathbf{W}$, computed in training phase, to transition the given noisy image in and out of the ICA domain. A major advantage of ICA domain is that a set of adaptive basis functions can be obtained from the image alone. The sparse code shrinkage is employed for denoising in the ICA domain.

3.2.1. Transitioning In and Out of the ICA Domain. As described above, processing in ICA domain allows stating of a set of adaptive basis functions from the image alone. Windowing approach is employed where an image sample is processed and replaced. An image sample $\mathbf{x}$ of size 8×8 is first projected in the ICA domain by multiplying it with $\mathbf{W}$. We denote the projection of the image in ICA domain as $\mathbf{s}$. For transitioning out of the ICA domain, $\mathbf{s}$ is to be multiplied by $\mathbf{W}^{-1}$. However, $\mathbf{W}$ is orthogonal, that is, $\mathbf{W}^{-1} = \mathbf{W}^T$; therefore, $\mathbf{s}$ is simply multiplied by $\mathbf{W}^T$. For computational purpose, $\mathbf{x}$ is reshaped to 64×1 while entering and again 8×8 while exiting ICA domain.

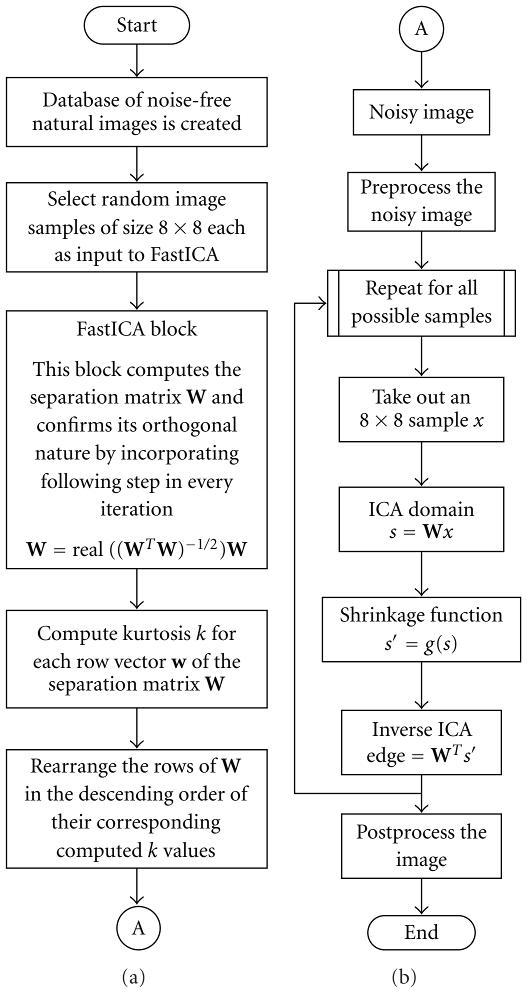

Start
Database of noise-free natural images is created
Select random image samples of size 8 × 8 each as input to FastICA
FastICA block
This block computes the separation matrix **W** and confirms its orthogonal nature by incorporating following step in every iteration
$\mathbf{W} = \text{real}\,((\mathbf{W}^T\mathbf{W})^{-1/2})\mathbf{W}$
Compute kurtosis k for each row vector **w** of the separation matrix **W**
Rearrange the rows of **W** in the descending order of their corresponding computed k values
A
(a)
A
Noisy image
Preprocess the noisy image
Repeat for all possible samples
Take out an 8 × 8 sample x
ICA domain $s = \mathbf{W}x$
Shrinkage function $s' = g(s)$
Inverse ICA edge = $\mathbf{W}^T s'$
Postprocess the image
End
(b)

Figure 3: Flowchart of the proposed approach with (a) training phase for estimating the sparse matrix and (b) edge-detection in the noisy gray-scale image are shown.

3.2.2. Denoising. Recovering the original image from a noisy image is complex. This work, therefore, aims at estimating an image, which is significantly better than the noisy image.

Probability densities encountered in image denoising can be classified as (i) mildly sparse and (ii) strongly sparse [7]. In this work, we employ the strongly sparse density, which, for any s_i in $\mathbf{s}$, is given by [21, 22]

$$p(s_i) = \frac{(\alpha+2)[\alpha(\alpha+1)/2]^{(\alpha/2+1)}}{[(\alpha(\alpha+1)/2)+|s_i/d|]^{(\alpha+3)}}. \qquad (6)$$

Here, d is the standard deviation, and α controls the sparseness. We first estimate $p(0)$ and d, using which, α is evaluated as

$$\alpha = \frac{2-k+\sqrt{k(k+4)}}{2k-1}, \qquad (7)$$

where

$$k = d^2 p(0)^2, \qquad d = \sqrt{E\{{s_i}^2\}}. \qquad (8)$$

Next, the maximum a posteriori (MAP) estimation of the density model (6) is found using the shrinkage function given by

$$g(s_i) = \frac{1}{2d}\,\text{sign}(s_i)m', \qquad (9)$$

where

$$m' = \max\left(0, \frac{|s_i|-ad}{2} + \frac{1}{2}\sqrt{(s_i+ad)^2 - 4\sigma^2(\alpha+3)}\right), \qquad (10)$$

$$a = \sqrt{\frac{\alpha(\alpha+1)}{2}}. \qquad (11)$$

In (10), σ represents noise level. Using the above equations, the value of $p(0)$ is approximated to be 0.707.

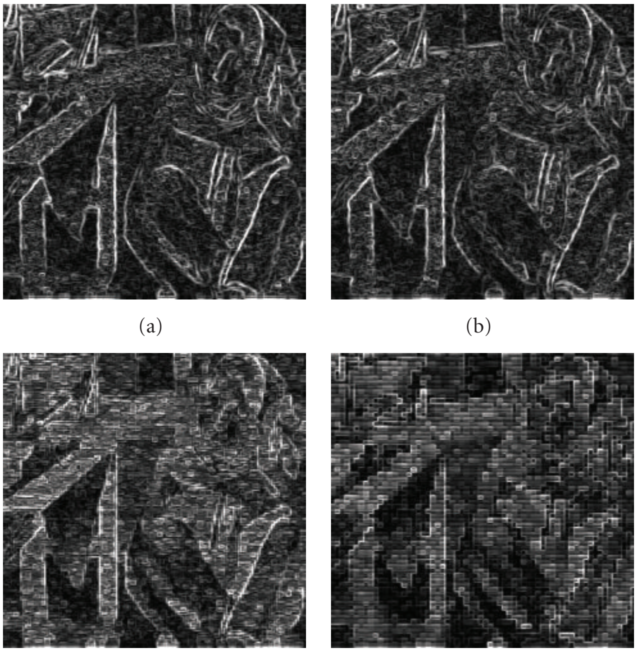
3.2.3. Discussion. The edge-detection phase implemented in this work is based on the sparse code shrinkage algorithm [8], where a sample is initially transformed into the ICA domain using **W**. Thereafter, these components are shrunk using the shrinkage function given in (9), and the sample is then taken out of the ICA domain using $\mathbf{W}^T$. Finally, Canny edge-detection algorithm [1] is applied to the image.

4. Illustration

The proposed approach is applied to both gray-scale and colored images, and the results obtained are compared with those obtained using Canny and mean filters. We have also studied the effects of step size as well as noise levels on the performance of our approach.

4.1. Training. For the training phase, a set of images are chosen from an image database [19] to create a local database (see Figure 2). Using the algorithm given in Figure 3(a), orthogonalized matrix **W** is estimated. Training phase can be accomplished with any set of noise-free images. Noise-free images for the image on hand are not required, since matrix **W** remains same for all known/unknown images [20].

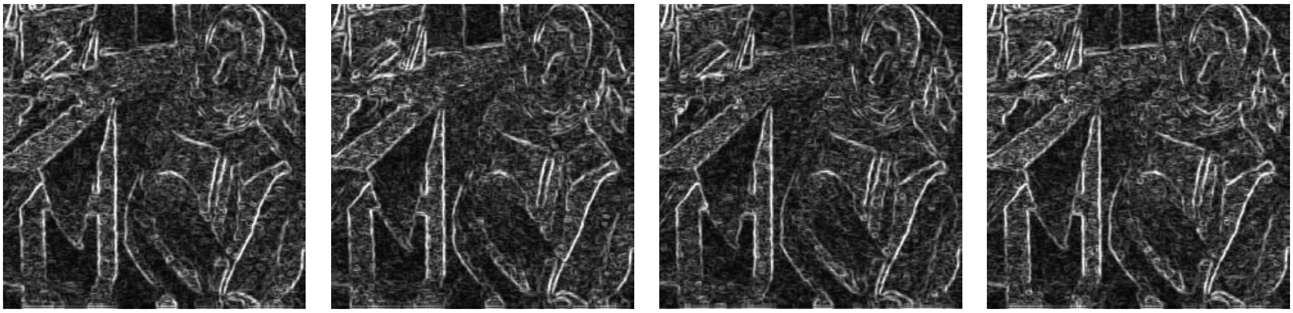
4.2. Effects of Noise and Step Size. Before proceeding to application of the proposed approach to gray-scale and colored images, we consider the effects of step size and noise levels on edge-detection. Step size is the distance between two successive image samples (i.e., sample windows). To study the effects of step size, the proposed approach is applied to an image with various step sizes (see Figure 4). As expected, the test image becomes increasingly susceptible to noise with increase in step size. However, increasing the step size improves the computational speed. As such, it is crucial to select a step size, which offers a compromise between the speed and immunity to noise of the proposed approach. For instance, for the image in Figure 4, a maximum step size of 4 can be used (for a step size of 8, the edges become indistinguishable). Processing of image is 16 times faster with step size 4 compared to that of step size 1.



(a) (b)

(c) (d)

Figure 4: Proposed approach applied to the standard Barbara image with 3 dB Gaussian noise. The step sizes chosen are (a) 1, (b) 2, (c) 4, and (d) 8, respectively.



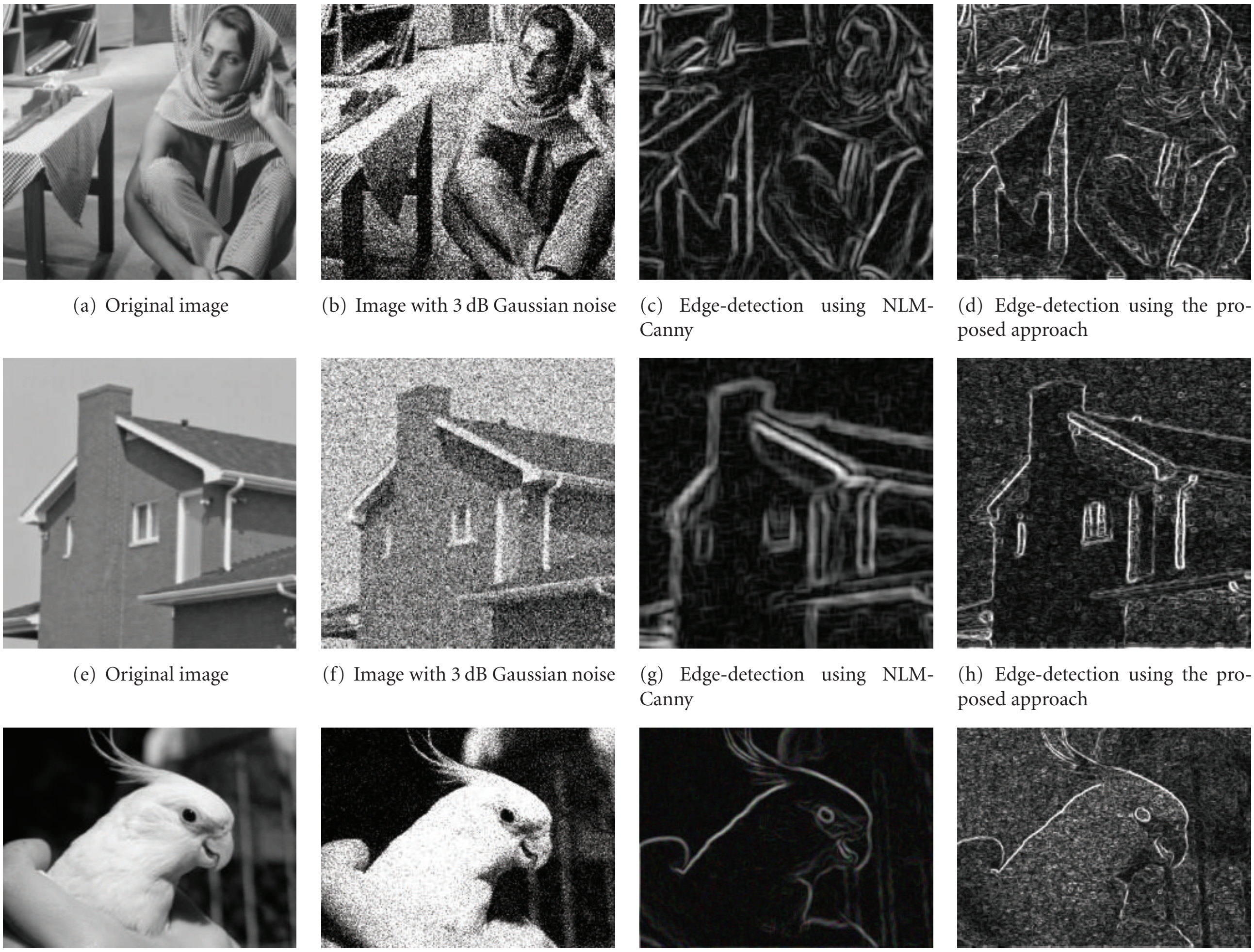
(a) 1 dB (b) 2 dB (c) 5 dB (d) 6 dB

Figure 5: Proposed approach applied to the Barbara image with various noise levels and step size 1.

Figure 5 illustrates the results of the proposed approach on an image corrupted with different levels of white Gaussian noise. From the results obtained, it appears that the proposed approach gives satisfactory results for noise levels as high as 6 dB. It would be of interest to test the proposed approach with real data, that is, data for which noise-free images are unavailable. However, such results are difficult to obtain and comment upon. As such, performance of the algorithm for images artificially corrupted with noise is shown. Images not used during training are tested with the proposed approach.

4.3. Application to Gray-Scale Images. The proposed approach is applied to a number of gray-scale images, with step size of 1. For illustration, consider Figure 6. Here, three gray-scale images shown in Figures 6(a), 6(e), and 6(i) are tested with the proposed approach. Figures 6(b), 6(f), and 6(j) show the respective gray-scale images with artificially added 3 dB white Gaussian noise. nonlocal means filter [23] is used for denoising followed by Canny method for edge-detection, and the results obtained using this approach (NLMCanny) are depicted in Figures 6(c), 6(g), and 6(k). Finally, Figures 6(d), 6(h), and 6(l) show the results of the proposed approach. As can be seen from the images, due to high level of Gaussian noise (i.e., 3 dB) in the images, Canny method fails to provide satisfactory results. However, the proposed approach offers considerably better results.

4.4. Application to Colored Images. The proposed approach is extended to colored images. Matrix **W**, computed for gray scale images, is employed for the colored images as well.



(a) Original image

(b) Image with 3 dB Gaussian noise

(c) Edge-detection using NLM-Canny

(d) Edge-detection using the proposed approach

(e) Original image

(f) Image with 3 dB Gaussian noise

(g) Edge-detection using NLM-Canny

(h) Edge-detection using the proposed approach

(i) Original image

(j) Image with 3 dB Gaussian noise

(k) Edge-detection using NLM-Canny

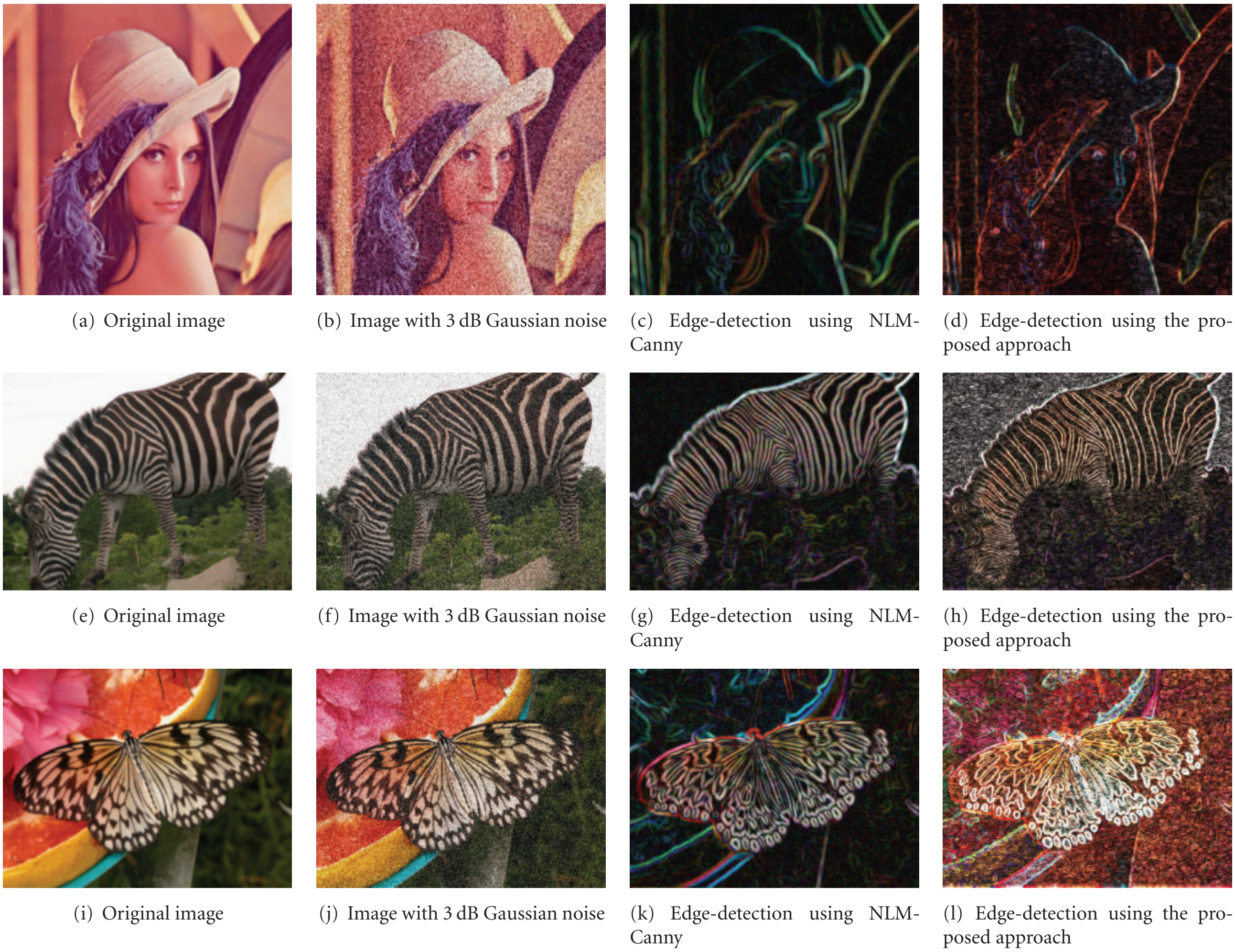
(l) Edge-detection using the proposed approach

FIGURE 6: Application of the proposed approach to gray-scale images with step size of 1.

White Gaussian noise is added separately and edge-detection phase is repeated individually for all three components R, G, and B. The results are given in Figure 7. Three colored images 7(a), 7(e), and 7(i) are tested with the proposed approach. Figures 7(b), 7(f), and 7(j) show the respective images with 3 dB white Gaussian noise added. Figures 7(c), 7(g), and 7(k) illustrate the results of NLMCanny applied to the respective noisy images, and the results of the proposed approach are shown in Figures 7(d), 7(h) and 7(l). Canny fails to provide satisfactory results because of the high noise level. However, our approach offers noticeably improved results.

4.5. Objective Evaluation of the Proposed Approach. Three important criterions must be fulfilled for the better performance of the edge-detection technique. (1) The edge detector should find all the real edges and not find any false edges. (2) The edges should be found in the correct place. (3) There should not be multiple edges found for a single edge [24]. Detailed subjective comparison for gray-scale as well as colored images is depicted in Figures 8(a), 8(c), and 8(e) in Figure 8 depict the NLMCanny results for gray-sale images. The respective proposed approach results are depicted in (b), (d), and (f) in Figure 8. Similarly. Figures 8(g), 8(i), and 8(k) depict the NLMCanny results for colored images, and the respective proposed approach results are depicted in Figures 8(h), 8(j), and 8(l). Apart from these subjective methods, there are objective methods which are used for the evaluation of edge detectors. However, these do not necessarily correlate with the human perception of the image quality. For example, an image with relatively lower error might look much worse than an image with relatively higher error. These objective measures are root-mean-square error (e_{RMS}), peak signal-to-noise ratio ($\mathrm{SNR}_{\mathrm{Peak}}$), signal-to-noise ratio ($\mathrm{SNR}_{\mathrm{RMS}}$), and Pratt's figure of merit (FOM) (refer to (12), (13), (14), and (15), resp.) [4, 24].

$$e_{\mathrm{RMS}} = \sqrt{\frac{1}{MN}\sum_{r=0}^{M-1}\sum_{c=0}^{N-1}\left[E(r,c) - O(r,c)\right]^2}, \qquad (12)$$


(a) Original image

(b) Image with 3 dB Gaussian noise

(c) Edge-detection using NLM-Canny

(d) Edge-detection using the proposed approach

(e) Original image

(f) Image with 3 dB Gaussian noise

(g) Edge-detection using NLM-Canny

(h) Edge-detection using the proposed approach

(i) Original image

(j) Image with 3 dB Gaussian noise

(k) Edge-detection using NLM-Canny

(l) Edge-detection using the proposed approach

FIGURE 7: Application of the proposed approach to colored images with step size of 1.

$$\mathrm{SNR}_{\mathrm{Peak}} = 10\log_{10}\frac{(L-1)^2}{(1/MN)\sum_{r=0}^{M-1}\sum_{c=0}^{N-1}\left[E(r,c)-O(r,c)\right]^2}, \tag{13}$$

$$\mathrm{SNR}_{\mathrm{RMS}} = \sqrt{\frac{\sum_{r=0}^{M-1}\sum_{c=0}^{N-1}\left[E(r,c)\right]^2}{\sum_{r=0}^{M-1}\sum_{c=0}^{N-1}\left[E(r,c)-O(r,c)\right]^2}}, \tag{14}$$

where $O(r,c)$ is the original image (one with added noise), $E(r,c)$ is the final image of size $M \times N$, and L is the number of gray levels, that is, 256. For colored images, objective measures are calculated for each channel and average is computed. Pratt's figure of merit is defined by [4]

$$\mathrm{FOM} = \frac{1}{I_N}\sum_{i=1}^{I_A}\frac{1}{1+\alpha d^2}, \tag{15}$$

$I_N = \max(I_A, I_I)$, where I_I and I_A represent the number of ideal and actual edge map points, α is a scaling constant, and d is the separation distance between actual and ideal edges. FOM is normalized such that FOM = 1 represents a perfectly detected edge. The value of α is set to 1/9. Table 1 shows the performance comparison between NLMCanny and proposed approach using the objective methods. It can be seen that the proposed approach outperforms NLMCanny since values for $\mathrm{SNR}_{\mathrm{RMS}}$, $\mathrm{SNR}_{\mathrm{Peak}}$, e_{RMS}, and FOM for the proposed approach are higher.

5. Conclusion

A new approach based on ICA has been proposed for edge-detection in noisy images and has been applied to both gray-scale and colored images. Compared to the NLMCanny method, that is, Canny coupled with a state-of-the-art nonlocal means filter, the proposed approach has yielded considerably improved results. Several step sizes have been examined, and the results for the same have been presented. Moreover, the proposed approach exhibits higher immunity to noise levels. Since all the natural images contain same type of statistics, shrinkage algorithm parameters can be computed from any set of noise-free images for denoising an image. As such, the proposed approach is very robust in the noisy environment. Results for noise levels of 1 dB, 2 dB, 5 dB, 6 dB, and so forth have been depicted. The proposed approach allows step sizes as high as 4 to be employed,

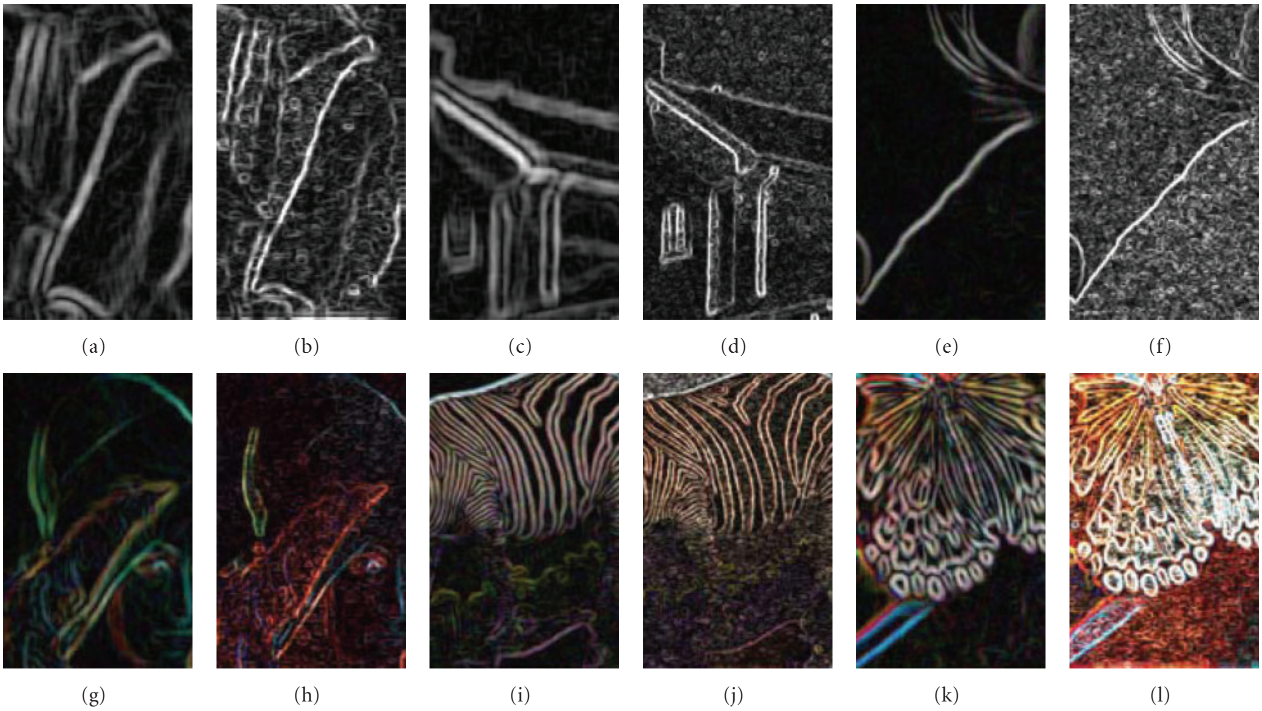


Figure 8: Detailed edge comparison for gray-scale and colored images, where (a), (c), (e), (g), (i), and (k) are NLM Canny results and (b), (d), (f), (h), (j), and (l) are the proposed approach results.

Table 1: Objective evaluation of the proposed approach.

Image	SNR_{RMS}		SNR_{Peak}		e_{RMS}		FOM	
	NLMCanny	Proposed	NLMCanny	Proposed	NLMCanny	Proposed	NLMCanny	Proposed
Gray-scale images								
Barbara	0.240	0.792	54.719	56.481	0.4084	0.4491	0.5256	0.6164
House	0.553	0.806	53.272	56.568	0.4185	0.4594	0.4820	0.7830
Bird	0.172	0.567	54.363	56.205	0.3880	0.3947	0.6714	0.6926
Colored images								
Lena	0.226	0.635	54.213	56.032	0.4010	0.4691	0.7142	0.7470
Zebra	0.290	0.564	55.094	55.379	0.4488	0.4582	0.7322	0.7511
Butterfly	0.299	0.640	55.814	56.375	0.3649	0.3981	0.6750	0.6840

thereby speedingup its execution on the computer. Change in step size from 1 to 4 increases the speed 16 times. Step size of 4 has been shown to produce acceptable results. On a 2.47 GHz PC equipped with 4 GB of RAM, NLMCanny took 238.1 seconds to process gray-scale Barbara image. The same image took 448 seconds using the proposed approach with the step size of 1 and 32.4 seconds with the step size of 4. Superiority of the proposed approach has been validated using both subjective as well as objective measures. As a continuation of this work, we plan to test the approach with images in other research areas (e.g., medical images). We also plan to test it on the images corrupted with other types of noises.

References

[1] J. Canny, "A computation approach to edge-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice-Hall, Upper Saddle River, NJ, USA, 3rd edition, 2008.

[3] M. Pathegama and Ö. Göl, "Edge-based image segmentation," *World Academy of Science, Engineering and Technology*, vol. 33, no. 2, pp. 164–167, 2005.

[4] W. K. Pratt, *Digital Image Processing*, John Wiley & Sons, Hoboken, NJ, USA, 2007.

[5] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Computation*, vol. 9, no. 7, pp. 1483–1492, 1997.

[6] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[7] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[8] X. Han, S. Dai, J. Li, and G. Xia, "Edge-detection algorithm based on ICA-domain shrinkage in noisy images," *Science in China F*, vol. 51, no. 9, pp. 1349–1359, 2008.

[9] Y. W. Chen, X. Y. Zeng, and H. Lu, "Edge-detection and texture segmentation based on independent component analysis," in *Proceedings of the 16th International Conference on Pattern Recognition*, pp. 351–354, Quebec, Canada, August 2002.

[10] S. Hornillo-Mellado, R. Martín-Clemente, J. I. Acha, and C. G. Puntonet, "Application of independent component analysis to edge-detection and watermarking," *Lecture Notes in Computer Science*, vol. 2687, pp. 273–280, 2003.

[11] X. H. Han, Y. W. Chen, and Z. Nakao, "Robust edge-detection by independent component analysis in noisy images," *IEICE Transactions on Information and Systems*, vol. E87-D, no. 9, pp. 2204–2211, 2004.

[12] P. Hoyer, *Independent component analysis in image denoising*, M.S. thesis, Helsinki University of Technology, 1999.

[13] P. Comon, "Independent component analysis, a new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[14] C. Jutten and J. Herault, "Blind separation of sources—part I: an adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, no. 1, pp. 1–10, 1991.

[15] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[16] J. Shlens, *A Tutorial on Principal Component Analysis*, Center for Neural Science, New York University, New York, NY, USA, 2005.

[17] Helsinki University of Technology, "The FastICA Package for MATLAB," 2005, http://research.ics.tkk.fi/ica/fastica/.

[18] J. Stone, *Independent Component Analysis: A Tutorial*, MIT Press, Cambridge, Mass, USA, 2004.

[19] A. Olmos and F. A. A. Kingdom, "McGill calibrated colour image database," 2004, http://tabby.vision.mcgill.ca/.

[20] A. J. Bell and T. J. Sejnowski, "The 'independent components' of natural scenes are edge filters," *Vision Research*, vol. 37, no. 23, pp. 3327–3338, 1997.

[21] A. Hyvärinen, "Sparse code shrinkage: denoising of non-gaussian data by maximum likelihood estimation," *Neural Computation*, vol. 11, no. 7, pp. 1739–1768, 1999.

[22] A. Hyvärinen, P. O. Hoyer, and E. Oja, "Image denoising by sparse code shrinkage," in *Intelligent Signal Processing*, IEEE Press, New York, NY, USA, 2000.

[23] A. Buades, B. Coll, and J. M. Morel, "A non local algorithm for image denoising," *IEEE Computer Vision & Pattern Recognition*, vol. 2, pp. 60–65, 2005.

[24] M. Roushdy, "Comparative study of edge-detection algorithms applying on the gray scale noisy image using morphological filter," *GVIP Journal*, vol. 6, no. 4, pp. 17–23, 2006.

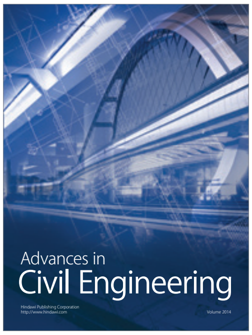
Journal of
Engineering

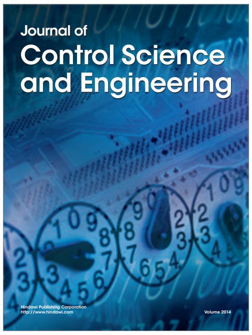
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

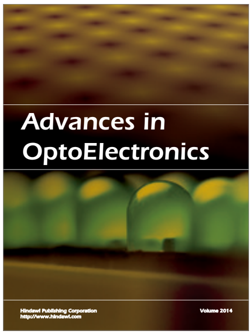
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

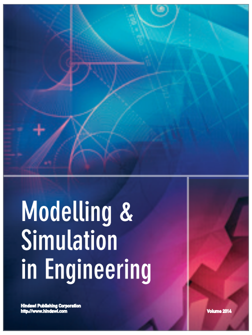
Journal of
Robotics

Journal of
Electrical and Computer Engineering

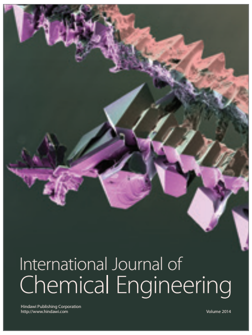

Advances in
OptoElectronics

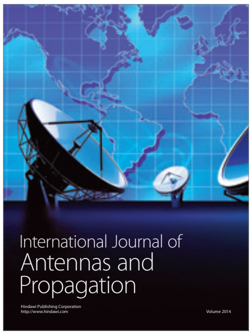
VLSI Design

International Journal of
Navigation and Observation

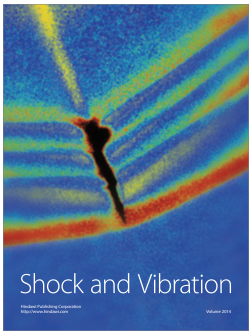

Modelling & Simulation in Engineering

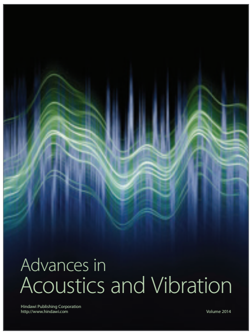
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration