Image Anal Stereol 2022;41:217-227 doi: 105566/ias.2812
Original Research Paper

FEATURE EXTRACTION FOR PATCH MATCHING IN PATCH-BASED DENOISING METHODS

GUANG YI CHEN ✉ AND ADAM KRZYZAK

Department of Computer Science and Software Engineering, Concordia University, Montreal, Quebec, Canada H3G 1M8
e-mail: guangyi_chen@hotmail.com, krzyzak@cse.concordia.ca

(Received October 3, 2022; accepted November 15, 2022)

ABSTRACT

Patch-based image denoising is a popular topic in recent years. In existing works, the distance between two patches was calculated as their Euclidian distance. When the noise level is high, this approach may not be desirable in image denoising. In this paper, we propose to extract noise-robust feature vectors from image patches and match the image patches by their Euclidian distance of the feature vectors for grey scale image denoising. Our modification takes advantage of the fact that the mean of the Gaussian white noise is zero. For every patch in the noisy image, we use lines to divide the patch into two regions with equal area and we take the mean of the right region for each line. Hence, a number of features can be extracted. We use these extracted features to match the patches in the noisy image. By introducing feature-based patch matching, our method performs favourably for highly noisy images.

Keywords: Additive Gaussian white noise; image denoising; patch-based image denoising.

INTRODUCTION

Noise reduction of an image is a very important problem in several real-life applications. An acquired image may be distorted by a few factors, including noise corruption. For instance, there may exist thermal noise, shot noise, Poisson noise, impulse noise, salt-and-pepper noise, etc. in a noisy image. Additive Gaussian white noise is the most popular noise addressed in the literature. In this paper, we formulate this kind of noise in image B as (Sendur and Selesnick, 2002):

$$B = A + \sigma_n Z \tag{1}$$

where A is the noise-free image, σ_n is the noise standard deviation, and Z is the Gaussian white noise with N(0,1) distribution..

There are many methods in the literature for reducing this kind of noise. Sendur and Selesnick (2002) proposed a bivariate denoising method by employing parent-child relationship in the wavelet domain. Dabov *et al.* (2007) developed the block matching and 3D filtering (BM3D) method, which is currently the state-of-the-arts in image denoising. The bounded BM3D (BBM3D) (Chen and Wu, 2010) first partitions an image into multiple regions and identifies the boundaries between regions. It restricts block matching search within the region of the template block. Then, it prevents important geometric features such as edges from being removed by collaborative filtering in BM3D. It does partial block matching for different block coherent segments which belong to different regions. Instead of postulating a statistical model for the wavelet coefficients, the SURE-LET (Luisier *et al.*, 2007) directly parametrizes the denoising process as a sum of elementary nonlinear processes with unknown weights. Then, it minimizes an estimate of the mean square error between the clean image and the denoised one. Contrary to the custom in the literature, the wavelet coefficients in the SURELET are not considered random anymore. They therefore proposed an interscale orthonormal wavelet thresholding algorithm for image denoising. Donoho and Johnstone (1994) proposed ideal spatial adaptation by wavelet shrinkage. Chen and Kegl (2007) proposed an image denoising method by using complex Ridgelets. Chen *et al.* (2005a, 2005b) developed two image denoising methods by considering coefficient dependency in the wavelet domain. Cho *et al.* (2009) also proposed three denoising techniques within the same paper by considering neighbour dependency in the wavelet domain for image denoising. Cho and Bui (2005) studied multivariate statistical modeling for image denoising using wavelet transforms. Recently, there are a few new image denoising methods appeared in the literature. Fathi and Naghsh-Nilchi (2012) proposed an efficient image denoising method based on a new adaptive wavelet packet thresholding function. Chatterjee and Milanfar (2012) studied patch-based near-optimal image denoising. Rajwade *et al.* (2013) worked on image denoising using the higher order singular value decomposition. Motta *et al.* (2011) proposed the Discrete Universal DEnoiser (iDUDE) framework for grey scale image denoising. It enhances the basic DUDE scheme by incorporating statistical information. Miller and Kingsburg (2008) studied image denoising using derotated complex wavelet coefficients. Lebrun (2012) analysed and implemented the BM3D Image Denoising Method. Hou *et al.* (2011) also fine-tuned the parameters in the original BM3D and better denoising results were obtained. Buades (2005) proposed a non-local means algorithm (NL-Means) for image denoising. Kervrann and Boulanger (2006) developed optimal spatial adaptation for patch-based image denoising. Talebi and Milanfar (2014) studied global image denoising. Yue *et al.* (2015) proposed a new image denoising method by exploring External and Internal Correlations. Zhang *et al.* (2017) proposed a Gaussian denoiser for residual learning of deep CNN for Image Denoising.

In this paper, we propose to extract noise-robust feature vector from each patch and the patch similarities are based on the Euclidean distance between the feature vectors. In existing patch-based methods, the patch similarities are based on the Euclidean distance of the patches, so they are less effective when the noise level is very high. To overcome this problem, we use several lines to divide each patch into two regions with equal area and then calculate the mean of right region. Since additive Gaussian white noise has zero mean, our calculated mean from the right region should be a good feature for patch matching. We use these extracted mean features to align the image patches. Experimental results show that our feature-based patch matching compares favourably to existing patch-based denoising methods in heavily noisy environment. This paper is an extension to our conference paper (Chen *et al.* 2014) published in the Proceedings of the Tenth International Conference on Intelligent Computing (ICIC), 2014.

EXISTING PATCH-BASED DENOISING METHODS

There are a few patch-based methods for image denoising in the literature. The BM3D is for reducing the additive Gaussian white noise in an image. It is since an image has a locally sparse representation in the transformed domain. The enhancement of sparsity is realized by grouping similar 2D patches of the noisy image into 3D data arrays, called "groups". Collaborative filtering is then performed according to the following three steps: a) 3D transformation of a group. b) Shrinkage of the transformed coefficients. c) Inverse 3D transformation. Since the patches overlaps with each other, there are multiple estimates for each pixel. These estimates are aggregated to obtain denoised images. Because the grouping of similar patches is based on the Euclidean distance of two patches, the BM3D will be less effective when the noise level is very high.

Chatterjee and Milanfar (2012) developed a locally optimal Wiener-filter-based method and extended it to take advantage of patch redundancy to improve the denoising performance. Their denoising method does not require parameter tuning and is practical, with the added benefit of a clean statistical motivation and analytical formulation. Their experimental results showed that their method produces results quite comparable with the state of the art.

Non-local means (NL-Means) (Buades 2005) employs the fact that natural images often contain patches in distant regions that are very similar to each other. It produces a denoised image by minimizing a penalty term on the average weighted distance between an image patch and all other patches in the image, where the weights are decreasing functions of the squared difference between the intensity values in the patches.

Kervrann and Boulanger (2006) developed a novel adaptive denoising algorithm where patch-based weights and variable window sizes are jointly utilized. An advantage of the method is that internal parameters can be easily chosen and are relatively stable. The method can denoise both piecewise-smooth and textured natural images since they contain enough redundancy. The performance of our algorithm is very close, and in some cases even surpasses, to that of the already published denoising methods.

Talebi and Milanfar (2014) developed a method for truly global filtering where each pixel is estimated from all pixels in the image. They gave a statistical analysis of their proposed global filter, based on a spectral decomposition of its corresponding operator, and they studied the effect of truncation of this spectral decomposition. In addition, they derived an approximation to the spectral (principal) components using the Nyström extension. Using these, they demonstrated that this global filter can be implemented efficiently by sampling a small percentage of the pixels in the image. Experiments demonstrated that their strategy could effectively globalize any existing denoising filters to estimate each pixel using all pixels in the image, hence improving upon the best patch-based methods.

PROPOSED PATCH-MATCHING TECHNIQUE

In this section, we propose to extract noise-robust features from each patch and then use the feature vectors to determine the similarity between two patches. Since the noise level is not known in a noisy image, we estimate it according to equation (2) given below. Instead of scalar wavelet transform, we perform dual-tree complex wavelet transform (DTCWT) (Kingsbury 2001) to the noisy image. According to Donoho and Johnstone (1994), the noise standard σ_n deviation can be approximated as:

$$\sigma_n = \frac{median(|y_{1i}|)}{0.6745}, \ y_{1i} \in \text{subband } HH_1. \qquad (2)$$

where HH1 is the finest scale of wavelet coefficient subband. Since we only need to perform the DTCWT transform on the noisy image for one decomposition scale to estimate σ_n ,it makes our estimation of σ_n very fast.

We know that the mean of Gaussian white noise is zero, so we divide each image patch into two regions with equal area and then calculate the mean of the right region. We choose several lines by connecting two points systematically so that a moderate number of features can be extracted from the image patches. Let $k \in [1, K]$, where $K{\times}K$ is the size of the image patch. We choose the lines passing through the following two points as in Case 1 or Case 2 (see Fig. 1 for an illustration):

Case 1: (1,k) and (K,K-k) (3)

Case 2: (k,1) and (K-k,K) (4)

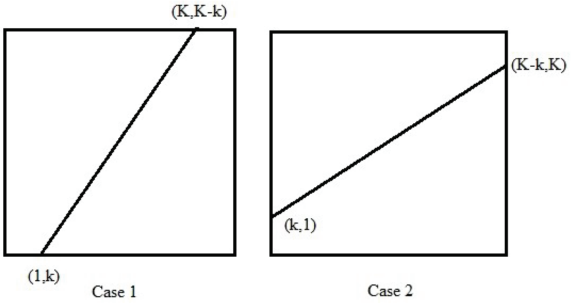

Fig. 1. *An illustration of lines that divide the patch into two regions with equal area.*

The total number of features for each patch is then *2K+1* whereas the patch size is *K×K*. It is easy to know that these lines divide the image patches into two regions with equal area. We then take the mean of the right region for each line and use the extracted mean features to match the patches. The closest patches should have the smallest distance in these features. Figs. 2 and 3 show the noise-free patch with 8×8 and 16×16 pixels, its noisy patch (σ_n=140), the extracted features from both the noise-free and noisy patches, and the difference between the noise-free and noisy features. The horizontal axes of the two lower sub-figures are the feature size, and the vertical axes represent the extracted features (lower-left) and the difference of features between the noise-free and noisy patches (lower-right). Our extracted features are very robust to Gaussian white noise because the difference between the extracted features is very small, compared to the magnitude of the extracted features.

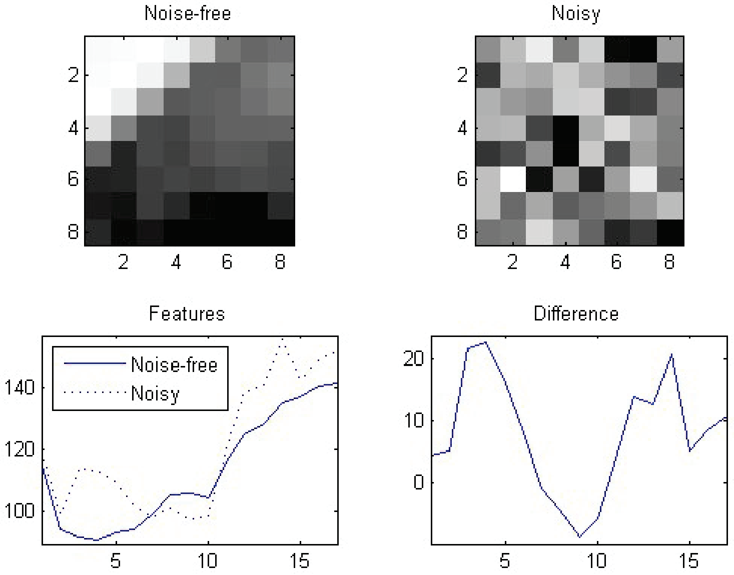

Fig. 2. *The noise-free patch with 8×8 pixels, its noisy patch (σ_n =140), the extracted features from both patches, and the difference between the noise-free and noisy features. The horizontal axes of the two lower sub-figures are the feature size, and the vertical axes represent the extracted features (lower-left) and difference of features between the noise-free and noisy patches (lower-right). It can be seen that our extracted features are robust to additive Gaussian white noise.*

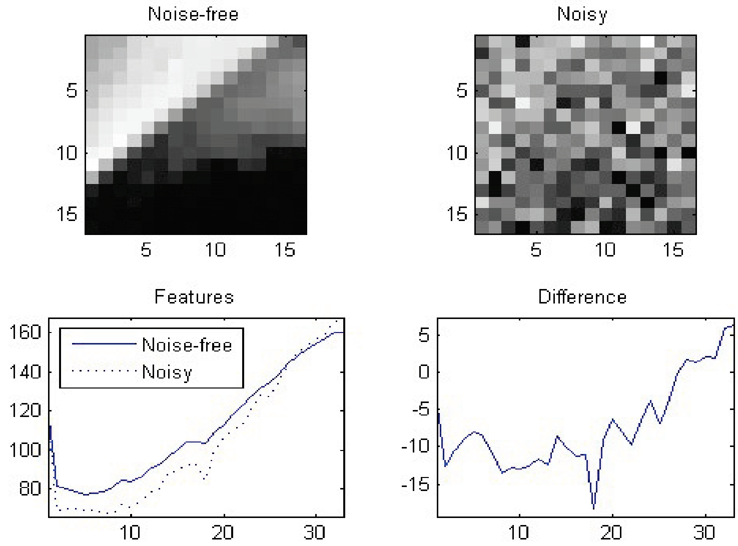

Fig. 3. *The noise-free patch with 16×16 pixels, its noisy patch (σ_n=140), the extracted features from both patches, and the difference between the noise-free and noisy features. The horizontal axes of the two lower sub-figures are the feature size, and the vertical axes represent the extracted features (lower-left) and difference of features between the noise-free and noisy patches (lower-right). It can be seen that our extracted features are robust to additive Gaussian white noise.*

Let us assume that every pixel in the noisy grey scale image is represented by L bits. Then, the pixel values should be in the range of 0 and *2^L-1*. We list the steps of our feature extraction and patch matching for image denoising as follows:

Step 1. Perform dual-tree complex wavelet transform (DTCWT) to noisy image B for one decomposition scale and denote the transformed image in the DTCWT domain as C. This transform is approximately shift invariant, which is very important in image denoising.

Step 2. Estimate the noise standard deviation σ_n from the transformed image C according to equation (2), given above.

Step 3. Compute threshold
$thr = 0.47 \times (2^L - 1)$

Step 4. If σ_n >thr, Then

Feature extraction from noisy patches:

1) Use a number of lines to divide each patch into two regions with equal area.
2) Calculate the mean of the right region and denote the extracted mean features as Fk (k=1, 2, …, 2K). Here the patch size is K×K pixels.
3) Use these extracted features to match the patches in the noisy image by calculating the Euclidian distance between two feature vectors.

Else

Use standard patch matching technique to calculate the patch distance.

End

Step 5. Follow the steps of existing denoising methods (e.g., NL-Means or BM3D) without any modifications.

Procedure 1:

Input: Patch size K by K.
Output: S+[][] of size 2K-1 by K (K+1)+1,
S-[][] of size 2K-1 by 4K+1.

```
// Initialization
for (int m=0; m< 2*K-1; m++) {
    for (int k=0; k< K*(K+1)+1; k++) {
        S+[m][k]=-1;
   }
   for (int k=0; k< 4*K+1; k++) {
        S-[m][k]=-1;
   }
}

// The first line
int k=0;
 for (int x=0: x<K; x++) {
      for (int y=x; y<K; y++) {
          S+[0][k] = x; S+[0][k+1] = y; k=k+2;
      }
  }

  // Other lines
   for (int i=0; i< K; i++)  { x1[i]=i; }
   for (k=1; k< K; k++) {
       for (int i=0; i< K; i++) {
          Find the intersection (x2[i],i) of the k-
th
          line with the horizontal line y=i.
          int min=0, plus=0;
          if (x2[i]>x1[i]) {
             for (int j=x1[i]; j<x2[i]; j++) {
                S-[k][min] = j;     S-[k+K-1][min]
= i;
                S-[k][min+1] = i; S-[k+K-
1][min+1] = j;
                min=min+2;
             }
           }  else {
              for (int j=x2[i]; j<x1[i]; j++) {
                 S+[k][plus] = j;   S+[k+K-1][plus]
= i;
                 S+[k][plus+1]=i; S+[k+K-
1][plus+1]= j;
                 plus=plus+2;
              }
           }

          // Backup
          x1[i]=x2[i];
          }

    }
```

Procedure 2:

Input: Noisy patch B of size K by K,
S+[][] of size 2K-1 by K (K+1)+1,
S-[][] of size 2K-1 by 4K+1.
Output: Feature vector F[] of length 2K-1.

```
// Calculate the feature vector
for (int k=0; k<2*K-1; k++)  {
     if (k==0)
        F[k]=0.0;
      else if (k==K)
        F[k]=F[0];
      else
        F[k]=F[k-1];

     // Newly introduced pixels
     int plus=0;
     while (S+[k][plus] != -1) {
          int x= S+[k][plus];
          int y= S+[k][plus+1];
          F[k] = F[k]+B[x][y];
          plus=plus+2;
      }

      // Discarded pixels
      int minus=0;
      while (S-[k][minus] != -1) {
           int x= S-[k][minus];
           int y= S-[k][minus+1];
           F[k] = F[k]-B[x][y];
           minus=minus+2;
      }
}

// Mean
double cnt=K*K/2;
for (int k=0; k<2*K-1; k++) {
   F[k]=F[k]/cnt;

}
```

In existing denoising methods, the noise variance σ_n is a known parameter for the noisy image. We estimate it by using equation (2) in this paper, so our proposed technique in this paper can be applied to any noisy images if additive Gaussian white noise is present in the image.

A faster implementation of feature extraction can be detailed here. Let F_k be the k-th feature extracted from a patch so that it is the mean of the right half of the patch divided by a line passing through the two points in the above Case 1 or Case 2. Similarly, we know that F_{k+1} is the (k+1)-th extracted feature from the patch. By careful study, we find out that.

$$F_{k+1} = F_k + \sum_{(i,j)\in R_+(k)} B(i,j) - \sum_{(i,j)\in R_-(k)} B(i,j)\,, \qquad (5)$$

where B(i,j) is the input noisy image, the first summation is for those pixels *R+(k)* that are introduced in *Fk*$_{+1}$, but are not in F_k. The second summation is for those pixels *R-(k)* that are discarded from F_k. A clever implementation is to pre-compute the pixel locations *R+(k)* and *R-(k)* for each F_k and save them in a file. We can then load this file when we denoise an image. Since F_k can be calculated sequentially from *k=1* until *k=2K* without multiplication and division, it is fast to extract these features. We give the pseudo-code for our implementation of feature extraction in Procedure 1 and Procedure 2. Procedure 1 needs to be computed only once.

Note that this implementation is like Huang *et al.* (1979), where, for the moving window, one row/column is added and at the same time one row/column is discarded. However, our implementation differs from that method in that we rotate the lines passing through two points instead of moving the window horizontally and vertically as in Huang *et al.* (1979).

EXPERIMENTAL RESULTS

We conducted several experiments to demonstrate the suitability of our feature-based patch matching technique in this paper. We tested five grey scale images: Lena, Boat, Barbara, Cameraman and Peppers. All these five images are represented by 8-bit (L=8) pixels and so in the range of 0 and 255. We generate the noisy images by using equation (1). Fig. 4 displays these five images, which are frequently used in other denoising papers in the literature. We incorporate our feature-based patch matching technique into BM3D and compare it with the standard BM3D, the bounded BM3D (BBM3D), and the SURELET for image denoising. Tables 1-5 tabulate the peak signal to noise ratio (PSNR) of the BBM3D, SURELET, standard BM3D and our modified BM3D for these five images, respectively. The PSNR is defined as

$$PSNR(X,Y) = 10\log_{10}\left(\frac{M\times N\times(2^L-1)^2}{\sum_{i,j}(Y(i,j)-X(i,j))^2}\right) \qquad (6)$$

where L is the number of bits representing a pixel in the images, $M \times N$ is the number of pixels in the image, and *X* and *Y* are the noise-free and denoised images. Our modified BM3D is identical to BM3D for noise standard deviation σ$_n$≤thr, and our modified BM3D outperforms the SURELET, BBM3D, and standard BM3D methods for nearly all other cases. The only exception is in Tables 4 and 5, where the standard BM3D is better than our modified BM3D for σ_n=140. Figs. 5-9 show the noise-free, noisy (σ_n =220), and the denoised images by SURELET, BBM3D, standard BM3D, and our modified BM3D for the Lena, Boat, Barbara, Cameraman, and Peppers images, respectively. Our denoised images are closer to the noise-free images than the images generated by all other three methods in the experiments. The images obtained by BBM3D and standard BM3D do not have smooth regions as the noise-free image, but our modified BM3D does.

Table 1. *The PSNR of different denoising methods for image Lena (512×512) with additive Gaussian white noise. The best results are highlighted in bold font.*

σ$_n$	Noisy	SURELET	BBM3D	BM3D	Proposed
80	10.06	25.45	24.97	**26.82**	**26.82**
100	8.12	24.59	23.67	**25.76**	**25.76**
120	6.53	23.90	22.60	**24.89**	**24.89**
140	5.20	23.32	21.59	23.62	**24.10**
160	4.04	22.81	20.65	21.95	**23.47**
180	3.01	22.37	19.77	20.86	**22.91**
200	2.10	21.96	18.88	19.98	**22.37**
220	1.27	21.59	17.99	19.15	**21.87**
240	0.51	21.25	17.09	18.45	**21.41**

Table 2. *The PSNR of different denoising methods for image Boat (512×512) with additive Gaussian white noise. The best results are highlighted in bold font.*

σ$_n$	Noisy	SURELET	BBM3D	BM3D	Proposed
80	10.06	23.79	23.63	**24.74**	**24.74**
100	8.12	23.06	22.48	**23.88**	**23.88**
120	6.53	22.49	21.53	**23.16**	**23.16**
140	5.20	22.03	20.73	22.21	**22.43**
160	4.04	21.63	19.96	20.92	**21.93**
180	3.01	21.29	19.23	20.12	**21.49**
200	2.10	20.99	18.56	19.46	**21.06**
220	1.27	**20.71**	17.91	18.81	20.67
240	0.51	**20.46**	17.21	18.22	20.30

Table 3. *The PSNR of different denoising methods for image Barbara (512×512) with additive Gaussian white noise. The best results are highlighted in bold font.*

σ$_n$	Noisy	SURELET	BBM3D	BM3D	Proposed
80	10.06	22.35	22.36	**24.84**	**24.84**
100	8.12	21.79	21.46	**23.66**	**23.66**
120	6.53	21.32	20.70	**22.69**	**22.69**
140	5.20	20.93	19.97	21.31	**21.64**
160	4.04	20.58	19.29	20.14	**21.20**
180	3.01	20.28	18.62	19.45	**20.79**
200	2.10	20.00	17.91	18.78	**20.44**
220	1.27	19.74	17.20	18.16	**20.09**
240	0.51	19.51	16.51	17.63	**19.77**

Table 4. *The PSNR of different denoising methods for image Cameraman (256×256) with additive Gaussian white noise. The best results are highlighted in bold font.*

σ_n	Noisy	SURELET	BBM3D	BM3D	Proposed
80	10.06	22.15	22.66	**24.00**	**24.00**
100	8.12	21.31	21.46	**22.99**	**22.99**
120	6.53	20.63	20.58	**22.13**	**22.13**
140	5.20	20.05	19.82	**21.24**	21.17
160	4.04	19.55	19.20	20.04	**20.69**
180	3.01	19.09	18.62	19.40	**20.27**
200	2.10	18.67	18.13	18.99	**19.89**
220	1.27	18.29	17.62	18.34	**19.48**
240	0.51	17.95	17.06	17.89	**19.22**

Table 5. *The PSNR of different denoising methods for image Peppers (256×256) with additive Gaussian white noise. The best results are highlighted in bold font.*

σn	Noisy	SURELET	BBM3D	BM3D	Proposed
80	10.06	22.26	23.16	**24.38**	**24.38**
100	8.12	21.35	21.98	**23.33**	**23.33**
120	6.53	20.65	20.91	**22.34**	**22.34**
140	5.20	20.07	20.01	**21.24**	21.21
160	4.04	19.56	19.16	19.99	**20.72**
180	3.01	19.10	18.48	19.32	**20.27**
200	2.10	18.67	17.80	18.70	**19.87**
220	1.27	18.28	17.19	18.00	**19.52**
240	0.51	17.91	16.53	17.35	**19.19**

Fig. 4. *The five images used in our experiments (from left to right): Lena, Boat, Barbara, Cameraman and Peppers.*

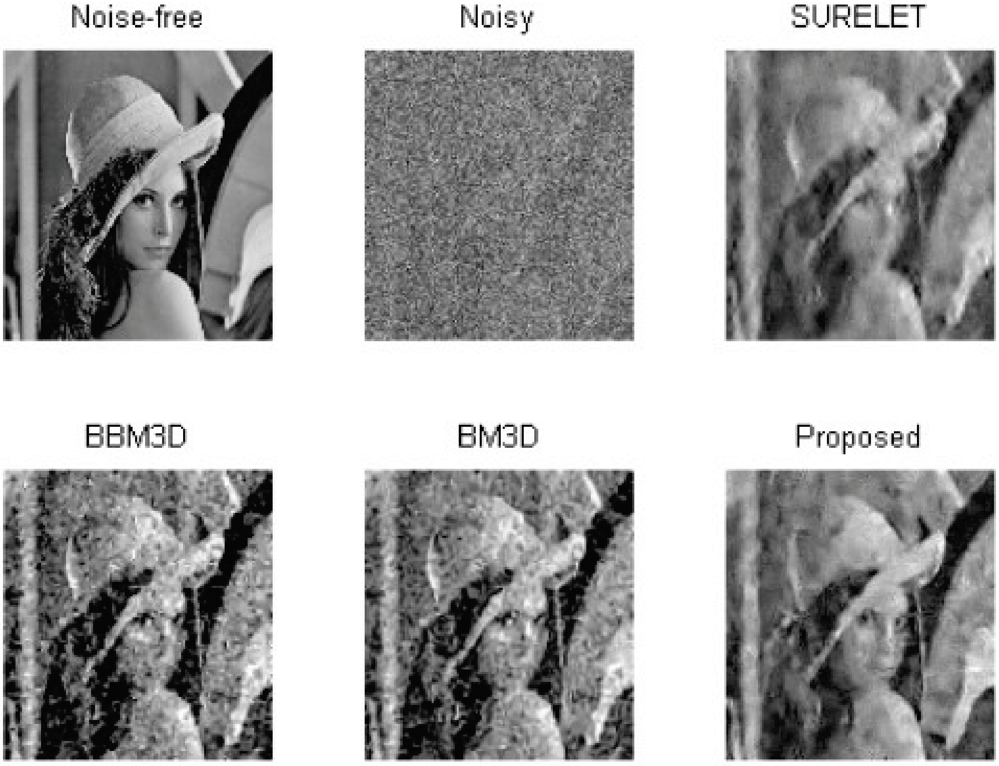

Fig. 5. *The noise-free, noisy (σ_n =220), and the denoised images by SURELET, BBM3D, BM3D, and our modified BM3D for the Lena image.*

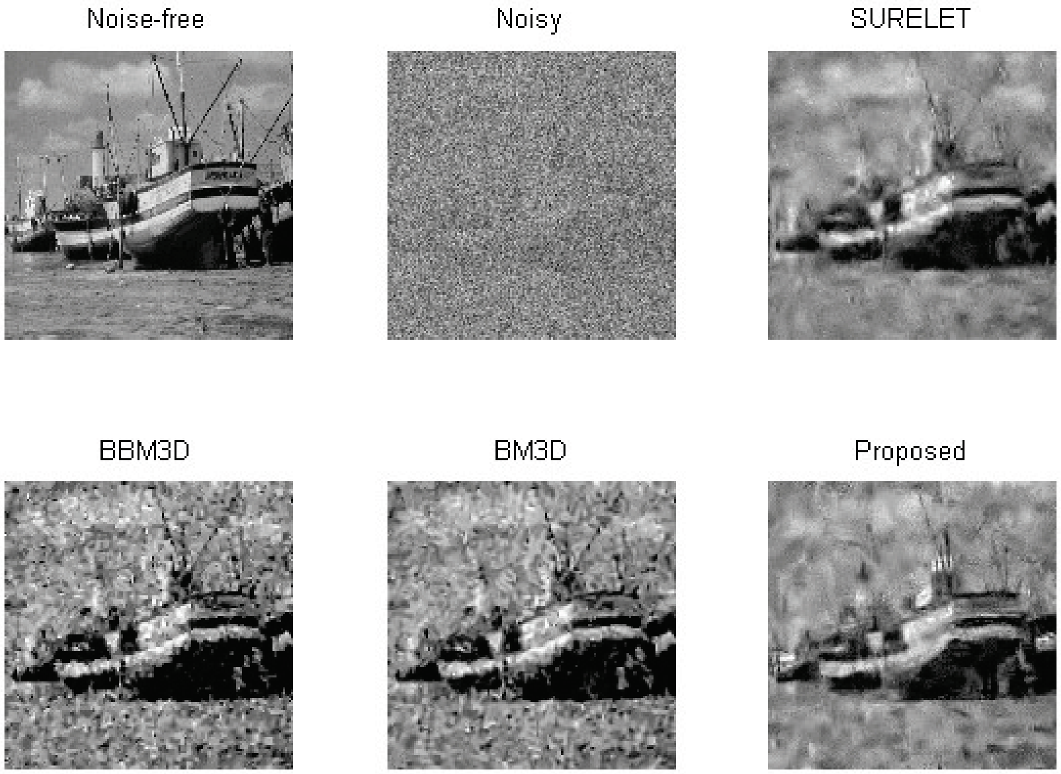

Fig. 6.*The noise-free, noisy (σ_n =220), and the denoised images by SURELET, BBM3D, BM3D, and our modified BM3D for the Boat image.*

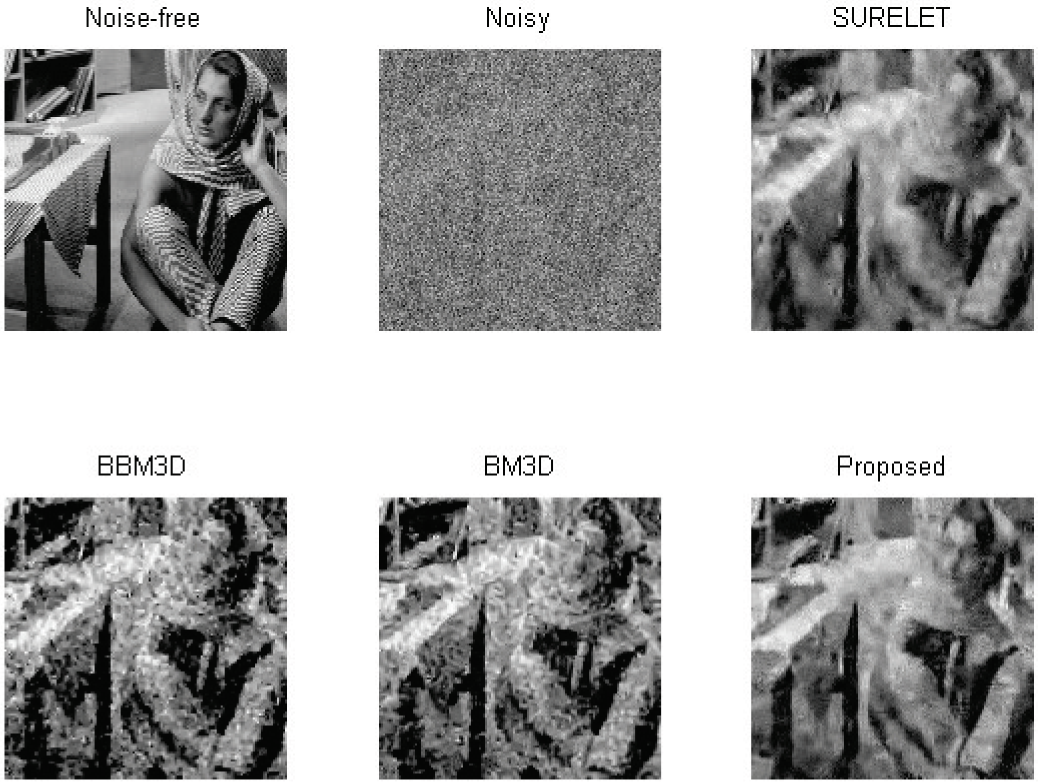

Fig. 7. *The noise-free, noisy (σ_n =220), and the denoised images by SURELET, BBM3D, BM3D, and our modified BM3D for the Barbara image.*

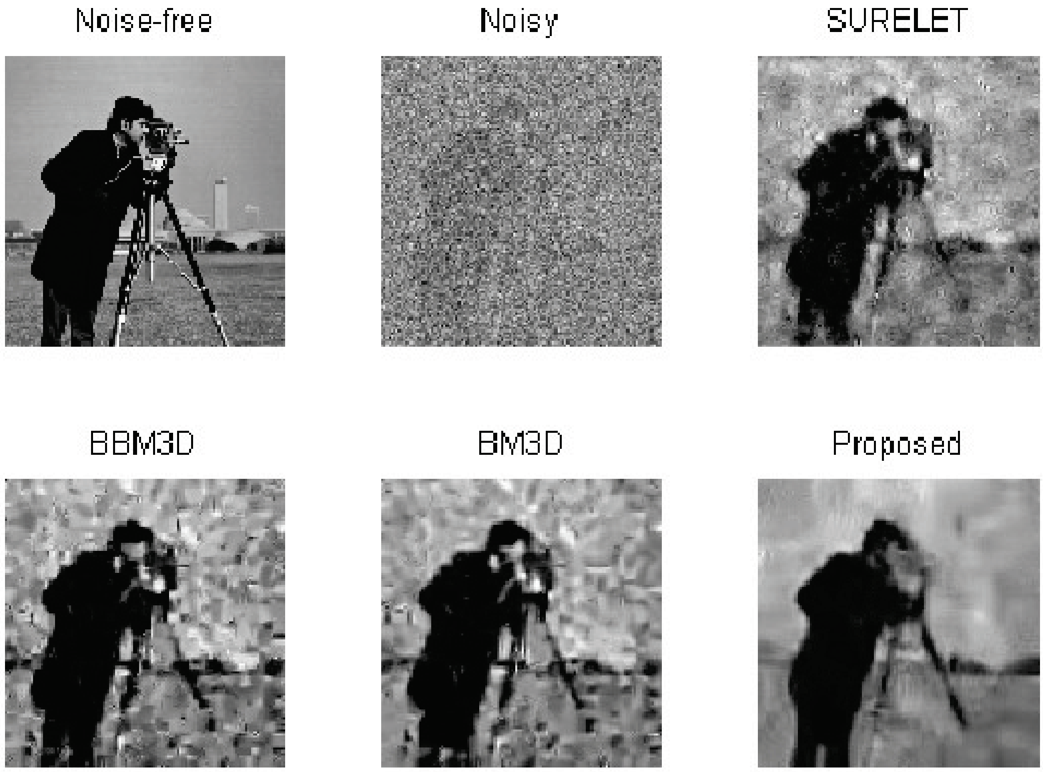

Fig. 8. *The noise-free, noisy (σ_n =220), and the denoised images by SURELET, BBM3D, BM3D, and our modified BM3D for the Cameraman image.*

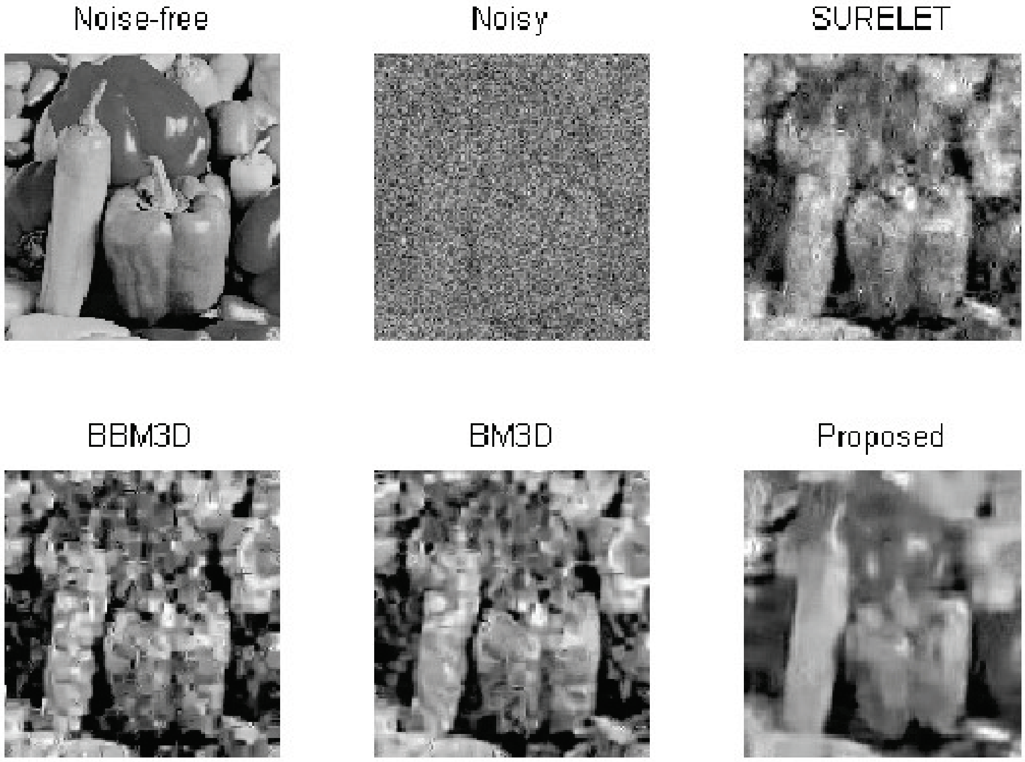

Fig. 9. *The noise-free, noisy (σ_n =220), and the denoised images by SURELET, BBM3D, BM3D, and our modified BM3D for the Peppers image.*

The major contribution of this paper is the following. Our proposed feature-based patch matching technique outperforms the standard methods when σ_n ≤thr, in terms of PSNR. Furthermore, the feature length of our extracted features from patches is shorter than the patch size. Our experiments show that our feature based BM3D compares favourably to the standard BM3D for denoising images in the heavily noisy environment. The only exception is for σ_n=140 and images Cameraman and Peppers, where the PSNRs of the standard BM3D for these two images are slightly better than those of our modified BM3D.

However, the difference in PSNR is marginal. Even though it is desirable to develop some methods for reducing other types of noise in an image, our modified BM3D might not work well in these cases, just as the standard BM3D.

CONCLUSIONS AND FUTURE WORKS

Images contaminated with noise introduce a critical problem in many real-life applications, including satellite imagery understanding, optical character recognition (OCR), historical document understanding, biometrics, and so forth. There are many kinds of noise present in a noisy image. In current literature, additive Gaussian white noise is the most popular topic discussed. We leave other kinds of noise to our future research.

In this paper, we have proposed a new technique for patch matching by extracting features from each patch and align the patches by using these feature vectors. The closest patches should have the smallest Euclidian distance in these feature vectors. Our experiments show that our feature-based patch matching technique produces better denoising results than standard patch matching methods under heavily noisy environment.

Future research will be conducted to deal with other types of noise in the noisy 1D signals, 2D images, and 3D videos. Also, we can extend our feature-based patch matching technique to reduce the noise in colour images. We believe that our proposed technique in this paper may also be applied to multi-spectral or hyper-spectral satellite imagery as well. Furthermore, we would like to align the image patches by some affine transform so that better denoising results can be obtained.

ACKNOWLEDGMENT

The authors thank the reviewer for his constructive comments. The authors would also like to thank the many authors for posting their denoising software on their websites. This research was supported by the research grant from the Natural Science and Engineering Research Council of Canada (NSERC).

REFERENCES

Antoni Buades (2005). A non-local algorithm for image denoising. Computer Vision and Pattern Recognition 2:60-65.

Chatterjee, P. and Milanfar, P. (2012). Patch-based near-optimal image denoising. IEEE T Image Process 21:1635-49.

Chen, G. Y., Bui, T. D. and Krzyzak, A. (2005a). Image denoising using neighbouring wavelet coefficients. Integr Comput-Aid E 12:99-107.

Chen, G. Y., Bui, T. D. and Krzyzak (2005b), A. Image denoising with neighbour dependency and customized wavelet and threshold. Pattern Recogn 38:115-24.

Chen, G. Y. and Kegl, B. (2007). Image denoising with complex ridgelets. Pattern Recogn 40:578-85.

Chen, G. Y., Xie, W. F. and Dai, S. (2014), Images denoising with feature extraction for patch matching in block matching and 3D filtering. Proceedings of the Tenth International Conference on Intelligent Computing (ICIC), Taiyuan, China.

Chen, Q. and Wu, D. (2010). Image denoising by bounded block matching and 3D filtering. Signal Process 90:2778-83.

Cho, D. and Bui, T. D. (2005). Multivariate statistical modeling for image denoising using wavelet transforms. Signal Process-Image 20:77-89.

Cho, D., Bui, T. D. and Chen, G. Y. (2009). Image denoising based on wavelet shrinkage using neighbour and level dependency. Intl J Wavelets, Multi 7:299-311.

Dabov, K., Foi, A., Katkovnik, V. and Egiazarian, K. (2007). Image denoising by sparse 3D transform-domain collaborative filtering, IEEE T Image Process. 16:2080-95.

Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. Biometrika 81:425-55.

Fathi, A. and Naghsh-Nilchi, A. R. (2012). Efficient image denoising method based on a new adaptive wavelet packet thresholding function. IEEE T Image Process 21:3981-90.

Hou, Y., Zhao, C., Yang, D. and Cheng, Y. (2011). Comments on "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering". IEEE T Image Process 20:268-70.

Huang, T. S., Yang, G. J. and Tang, G. Y. (1979). A fast two-dimensional median filtering algorithm. IEEE T Acoust Speech 27:13-8.

Kervrann, C. and Boulanger, J. (2006). Optimal spatial adaptation for patch-based image denoising. IEEE T Image Process 15:2866-78.

Kingsbury, N. G. (2001). Complex wavelets for shift invariant analysis and filtering of signals. Appl Comput Harmon Anal 10:234-53.

Lebrun, M. (2012). An Analysis and Implementation of the BM3D Image Denoising Method. Image Processing. On Line. http://dx.doi.org/10.5201/ipol.2012.l-bm3d.

Luisier, F., Blu, T. and Unser, M. (2007). A new SURE approach to image denoising: Interscale orthogonal wavelet thresholding, IEEE T Image Process 16:593-606.

Miller, M. and Kingsburg, N. (2008). Image denoising using derotated complex wavelet coefficients. IEEE T Image Process 17:1500-11.

Motta, G., Ordentlich, E., Ramirez, I., Seroussi, G. and Weinberger, M. J. (2011). The iDUDE framework for grayscale image denoising. IEEE T Image Process 20:1-21.

Rajwade, A., Rangarajan. A. and Banerjee, A. (2013). Image denoising using the higher order singular value decomposition. IEEE T Pattern Anal 35:849-62.

Sendur, L. and Selesnick, I, W. (2002). Bivariate shrinkage with local variance estimation. IEEE Signal Proc Let 9:438-41.

Talebi, H. and Milanfar, P. (2014). Global image denoising. IEEE T Image Process 23:755-68.

Yue, H., Sun, X., Yang, J. and Wu, F. (2015). Image Denoising by Exploring External and Internal Correlations. IEEE T Image Process 24:1967-82.

Zhang, K., Zuo, W., Chen, Y., Meng, D. and Zhang, L. (2017). Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. IEEE T Image Process, 26:3142-55.