Image Restoration Using Hybrid Features Improvement on Morphological Component Analysis

Der-Chang Tseng | Ru-Yin Wei | Ching-Ta Lu* | Ling-Ling Wang

Abstract—Images are generally corrupted by impulse noise during acquisition and transmission. Noise deteriorates the quality of images. To remove corruption noise, we propose a hybrid approach to restoring a random noise-corrupted image, including a block matching 3D (BM3D) method, an adaptive non-local mean (ANLM) scheme, and the K-singular value decomposition (K-SVD) algorithm. In the proposed method, we employ the morphological component analysis (MCA) to decompose an image into the texture, structure, and edge parts. Then, the BM3D method, ANLM scheme, and K-SVD algorithm are utilized to eliminate noise in the texture, structure, and edge parts of the image, respectively. Experimental results show that the proposed approach can effectively remove interference random noise in different parts; meanwhile, the deteriorated image is able to be reconstructed well.

Index Terms—Adaptive non-local mean (ANLM), block matching 3D (BM3D), image restoration, morphological component analysis (MCA), singular value decomposition (SVD).

1. Introduction

In recent years, communications and the Internet have been developed substantially, and images can be obtained through various consumer electronic products. Consumer demands for image quality have also increased. An image would be corrupted by two types of noise, including the Gaussian noise and salt-and-pepper noise. The Gaussian noise is generated by the random disturbance in a signal and causes the corrupted image to be unclear. The gray-level of a corrupted pixel can be any random value. It is a heavy task for the removal of random noise in an image.

Many methods for image noise removal have been proposed[1][11], the most popular of which are the median filter (MF) and a variety of filters based on MF. The adaptive median filter (AMF)[1] and weighted median filter (WMF)[2] are often used to remove background noise for a noisy image. In [3], directional-WMF, which modified the gray level of a noisy pixel by the weighted median of the pixels on a direction with minimum variations for image

*Corresponding author
Manuscript received 2018-08-12; revised 2019-02-17.
This work was supported by MOST under Grant No. 104-2221-E-468-007.
D.-C. Tseng is with the Department of Electronic Engineering, National Taipei University of Technology, Taipei 10608 (e-mail: dctseng@ntut.edu.tw).
R.-Y. Wei is with Quanta Computer Inc., Taipei 11167 (e-mail: s3623431@gmail.com).
C.-T. Lu and L.-L. Wang are with the Department of Information Communication, Asia University, Taichung 41354 (e-mail: Lucas1@ms26.hinet.net; ling@asia.edu.tw).
Publishing editor: Yu-Lian He
denoising, was proposed. This method only has four candidate directions. In [4], an improved version was proposed. The improvement is due to the increase of candidate directions and the exclusion of extreme pixels for the reconstruction of noisy pixels. Dabov et al. [5] proposed using sparse representations in the transform domain for image denoising. Collaborative filtering is performed to filter 3D data arrays, enabling random noise to be effectively removed. Deng and Liu (DL) [6] proposed an image denoising method combined with MF and the morphological component analysis (MCA) method. This method can effectively analyze the details of a noisy image, so interference noise can be effectively removed.

Although the method proposed in [6] can effectively analyze the details of the image, the denoising performance can be further improved. In the proposed method, we employ MCA to decompose an image into the texture, structure, and edge parts, as the DL method [6]. In turn, a block matching 3D (BM3D) scheme is used to eliminate noise in the image texture part. An adaptive non-local mean (ANLM) scheme is used to eliminate noise in the image structure part, and the K-singular value decomposition (K-SVD) algorithm is utilized to eliminate noise in the image edge part. Experimental results show that the proposed method can provide better performance than the DL method. Thus the performance of the DL method is improved.

The rest of this paper is organized as follows: Section 2 introduces the proposed image enhancement approach. Section 3 shows the experimental results. Finally, Section 4 draws conclusions.

2. Proposed Image Enhancement Approach

The flowchart of the proposed image denoising system is shown in Fig. 1. Initially, a noisy image is analyzed using the improved MCA algorithm. The image is decomposed into the image texture, structure, and edge parts. In turn, noise removal is performed for each different part. The image texture is denoised using the BM3D algorithm, while the structure portion is denoised using the ANLM method with an adaptive search window. The image edge portions are denoised using the K-SVD method. Finally, the three denoised portions are merged to produce a denoised image.

2.1. Improved MCA

Starck et al. [12, 13] proposed a signal separation method with sparse representations, and named it as MCA, which mainly uses human visual characteristics to classify image components, such as texture and structure. MCA assumes that there is a dictionary corresponding to the source signal, which can sparsely represent the source signal and emphasize that the dictionary can only sparsely represent the signal. The dictionary can be obtained in various transform domains, such as wavelet transform [14] and curvelet transform [15].

Assume that an image \( x \) is represented by a dictionary matrix \( D \) and consists of \( K \) sub-dictionaries. A noisy image \( y \) can be expressed by

\[
y = \sum_{k=1}^{K} x_k + n
\]

with
\[ x_k = D_k \alpha_k, k = 1, 2, \ldots, K. \] (2)

Subject to
\[
\min_{\alpha_1, \alpha_2, \ldots, \alpha_K} \sum_{k=1}^{K} |\alpha_k|^p
\] (3)
such that
\[
|y - \sum_{k=1}^{K} D_k \alpha_k|_2 \leq \tau
\] (4)

where \( \alpha_k \) is the sparse coefficient. \( x_k \) and \( n \) denote noise-free and interference noise, respectively. \( \tau \) is a distance threshold.

The detailed procedure for MCA is shown in Fig. 2\textsuperscript{[86]}. If the dictionary \( D \) is known, the orthogonal matching pursuit (OMP) algorithm\textsuperscript{[37]} can be used for the optimization of (3) and (4), which can be regarded as a dictionary learning process. The dictionary \( D \) requires to update step by step. On the contrary, the dictionary \( D \) is unknown. Learning procedures are needed for finding the dictionary \( D \). The dictionary learning method is to fill the dictionary matrix and calculate the sparse coefficient through the OMP algorithm by selecting some row vectors in a signal. The dictionary and coefficients are constantly updated alternately, and finally the best approximation representation of the original image is found. In this paper, we combine these two methods by giving a known dictionary as an initial state and updating the dictionary through a learning method. This combined method is named as the improved MCA algorithm and can be more adaptive to the signal than the original dictionary.

2.2. ANLM Filtering

An adaptive search window with a non-local mean (NLM) filter, that is ANLM, is employed to remove background noise and preserve more image edges and details. The NLM filter\textsuperscript{[7]} performs image denoising in a spatial domain filter. Considering the pixel variation in a neighbor region enables the NLM filter to preserve the image edge and detail information well.

A noisy image can be alternatively expressed by
\[ y(j) = x(j) + n(j), j \in I \] (5)

where \( j \) is the position index. \( I \) represents the pixel set of an image.

The gray-level of a noise pixel is replaced by a weighed value on the neighbor similar pixels, enabling noise pixels to be removed. The denoised pixel can be obtained by
\[
\hat{x}(i) = \sum_{j \in l} w(i, j) y(j), \forall i \in I, j \in l
\] (6)

where \( I \) denotes the neighborhood of the pixel centered at \( i \). \( w(i, j) \) represents the weight of the neighbor pixels, which can be computed by
\[ w(i, j) = \frac{1}{Z_i} e^{-\frac{\|Y_i - Y_j\|^2}{\sigma^2}} \]  
(7)

where \( Y(i) \) denotes a local window with a size of \( N \times N \). \( \sigma \) represents the standard deviation of noise. \( Z_i \) is a normalization factor, given as

\[ Z_i = \sum_j e^{-\frac{\|Y_i - Y_j\|^2}{\sigma^2}}. \]  
(8)

In order to define the window size for the NLM filter, an adaptive isotropic search window \(^{[10]}\) is utilized. The search window size is given as

\[
s_{\text{opt}}^i = \begin{cases} 
\text{search window } 9 \times 9 , & \text{if } \Delta \tilde{X} < T_1 \\
\text{search window } 15 \times 15 , & \text{if } T_1 \leq \Delta \tilde{X} \leq T_2 \\
\text{search window } 21 \times 21 , & \text{if } \Delta \tilde{X} > T_2 
\end{cases} \]  
(9)

where \( T_1 \) and \( T_2 \) denote the thresholds of the pixel variation. \( \Delta \tilde{X} \) represents the pixel variation, which can be computed by

\[ \Delta \tilde{X} = |\tilde{X}(i) - \bar{\tilde{X}}(i)| \]  
(10)

where \( \tilde{X}(i) \) denotes the filtered pixel obtained by (6) with a window size of \( 7 \times 7 \), the searching window size is \( 21 \times 21 \). \( \bar{\tilde{X}}(i) \) is the average gray level of the neighbor filtered pixels, which is computed by

\[ \bar{\tilde{X}}(i) = \frac{1}{M_1 \times M_2} \sum_{k} \sum_{l} \tilde{X}(k, l), \tilde{X}(k, l) \in I. \]  
(11)

In (9), the thresholds \( T_1 \) and \( T_2 \) can be obtained by

\[ T_1 = \frac{1}{M_1 \times M_2} \sum_k \tilde{X}(k) \]  
(12)

and

\[ T_2 = T_1 + \rho v \]  
(13)

where \( M_1 \) and \( M_2 \) denote the sizes of an image for the width and height, respectively. \( \rho \) is an empirical constant. \( v \) denotes the standard deviation of the filtered pixels, given as

\[ v = \left\{ \frac{1}{M_1 \times M_2} \sum_k [\tilde{X}(k) - T_1]^2 \right\}^{1/2}. \]  
(14)

The denoised pixel in the structure portion is obtained by

\[ \tilde{X}_s = \sum_{j=\text{s}} w(i, j) y(j). \]  
(15)

### 2.3. BM3D Filtering

The BM3D algorithm\(^{[5]}\) is applied to the image texture part for noise removal. The flowchart of the BM3D algorithm is shown in Fig. 3.

As shown in Fig. 3, the BM3D algorithm uses NLM mentioned in subsection 2.1 to find the similar blocks at a specific search window. According to the Euclidean distance, similar 2D blocks are searched, and the 2D blocks are superimposed on each other into the 3D space. Hence 3D transforming to the frequency domain and hard
thresholding are performed to remove interference noise. The features are converted back to the original image. A basic estimate is obtained. Finding similar blocks and converting to the frequency domain with Wiener filtering are further performed as the second stage. A denoised image is obtained.

The basic estimate in the first stage can be obtained by [6]:
\[
\hat{y}(x) = \frac{\sum \sum w_{\sigma}^b Y_{\sigma}^b(x)}{\sum \sum w_{\sigma}^b X_{\sigma}(x)}, \quad \forall x \in I
\]  
where \( X_{\sigma}(x) : I \to \{0, 1\} \) denotes the characteristic function of the square support of a block. \( w_{\sigma}^b \) denotes the weighting factor. \( Y_{\sigma}^b(x) \) represents a hard thresholded value, it is achieved by
\[
Y_{\sigma}^b = T_{3D}^{-1}(r(T_{3D}(I_{\sigma}))
\]
where \( I_{\sigma} \) denotes similar 3D blocks and \( T_{3D} \) denotes the operator of 3D transform. \( r \) represents the hard threshold, which is given as
\[
r = \begin{cases} 
2.7\sigma, & \text{if } \sigma \leq 40 \\
2.8\sigma, & \text{if } \sigma > 40.
\end{cases}
\]

In the second stage, Wiener filtering is performed for the removal of background noise. The gray level of a denoised pixel is computed by [5]:
\[
\tilde{y}(x) = \frac{\sum \sum w_{\sigma}^w Y_{\sigma}^w(x)}{\sum \sum w_{\sigma}^w X_{\sigma}(x)}, \quad \forall x \in I
\]
where \( w_{\sigma}^w \) denotes the weighting factor. \( Y_{\sigma}^w \) denotes a Wiener filtered value, it is computed by
\[
Y_{\sigma}^w = T_{3D}^{-1}(W_{\sigma}^w T_{3D}(I_{\sigma}))
\]
where \( I_{\sigma} \) denotes similar 3D blocks. \( W_{\sigma}^w \) represents the Wiener filter, it is expressed by [5]:
\[
W_{\sigma}^w = \frac{|T_{3D}(Y_{\sigma}^w)|^2}{|T_{3D}(Y_{\sigma}^w)|^2 + \sigma^2}.
\]
2.4. K-SVD Algorithm

The K-SVD algorithm\(^{[9]}\) is a method for adapting the dictionary to sparsely represent a signal. Given a signal, our task is to seek codewords to represent the signal from a dictionary and satisfy strict sparse signal constraints. This method is an alternate iterative algorithm for sparse coding. Based on the current dictionary, two steps of sparse coding and dictionary update are used to facilitate better adaptation of signals. The dictionary column update and the related sparse coefficient update are performed simultaneously, which can accelerate the convergence speed. The K-SVD algorithm has high flexibility and can adapt well to different signals. The detailed procedures of the K-SVD algorithm are shown in Fig. 4.

3. Experimental Results

The proposed method is based on a denoising method proposed by DL\(^{[3]}\) and attempts to improve the performance of the DL method. The DL method divides an image into the texture, structure, and edge parts for image denoising. In the DL method, the texture, structure, and edge parts are denoised using the wavelet-based method\(^{[6]}\), MF incorporated with the K-SVD algorithm, and K-SVD algorithm, respectively. By comparing the DL and proposed methods, both methods use the K-SVD algorithm for noise removal in the image edge part. While different methods are used for image denoising in the texture and structure parts. Accordingly, we will show the improvement of our method for the DL method in the image texture and structure parts.

Some objective measures are utilized to measure the performance of the proposed method, including peak signal-to-noise ratio (PSNR), mean-square-error (MSE), root-mean-square-error (RMSE), and mean-absolute-error (MAE). MSE and PSNR can be computed by

\[
\text{MSE} = \frac{1}{M_1 \times M_2} \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} (X_{ij} - \hat{X}_{ij})^2
\]  

(22)

\[
\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}} \text{ (dB)}.
\]  

(23)

The larger the PSNR value, the smaller the difference between the two image signals. The larger value of PSNR and the smaller value of MSE correspond to better image quality. RMSE and MAE are defined as

\[
\text{RMSE} = \left[ \frac{1}{M_1 \times M_2} \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} (X_{ij} - \hat{X}_{ij})^2 \right]^{1/2}
\]  

(24)

\[
\text{MAE} = \frac{\sum_{i=1}^{M_1} \sum_{j=1}^{M_2} |X_{ij} - \hat{X}_{ij}|}{M_1 \times M_2}.
\]  

(25)

The smaller values of RMSE and MAE correspond to better image quality.
3.1. Denoising of Image Texture

Table 1 shows the PSNR comparisons for the noise removal methods of different image textures. The proposed method is better than the DL method on average by about 3 dB and 6 dB when the noise densities are $\sigma = 15$ and $\sigma = 45$, respectively. The proposed method significantly improves the performance of the DL method for the image texture part in heavy noise corruption.

Table 2 shows the RMSE comparisons of the noise removal methods for different image texture parts. RMSE of the proposed method is less than that of the DL method about 3 and 10 in slight (noise density $\sigma = 15$) and heavy (noise density $\sigma = 45$) noise corruption. These results are consistent with the performance in terms of PSNR. Thus the proposed method outperforms the DL method for the restoration of texture parts. Table 3 shows the MAE comparisons of the noise removal methods for different image texture parts. The values of MAE of the proposed method are less than those of the DL method in all conditions. These results are consistent with Tables 1 and 2. Therefore, it is convinced that the proposed method significantly improves the performance of the DL method for the image texture.

Fig. 5 illustrates the image texture. By comparing Figs. 5 (a), (c), and (d), we can find that the texture denoised by the proposed approach (Fig. 5 (d)) is more similar to the original image texture (Fig. 5 (a)) than that produced by the DL method (Fig. 5 (c)). By observing the texture image denoised by the DL method (Fig. 5 (c)), some of the strip texture details at the position of the scarf disappear. On the contrary, the proposed method can well restore the image texture, in particular at the position of the scarf. From the performance presented in Tables 1 to 3 and Fig. 5, it is convinced that the proposed method can significantly improve the DL method for the restoration of the image texture.

![Image](image_url)

Fig. 5. Texture of the restored Barbara image: (a) noise-free image, (b) noisy image with noise density $\sigma = 25$, (c) texture of the denoised image using the DL method, and (d) texture of the denoised image using the proposed method.

<p>| Table 1: Performance comparisons in terms of PSNR for the image texture part |
|-----------------------------|------------------------|------------------------|------------------------|------------------------|</p>
<table>
<thead>
<tr>
<th>Image</th>
<th>Method</th>
<th>$\sigma = 15$</th>
<th>$\sigma = 25$</th>
<th>$\sigma = 35$</th>
<th>$\sigma = 45$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barbara</td>
<td>Noisy</td>
<td>25.88</td>
<td>20.43</td>
<td>17.52</td>
<td>15.31</td>
</tr>
<tr>
<td></td>
<td>DL</td>
<td>29.74</td>
<td>26.42</td>
<td>23.83</td>
<td>21.95</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>33.09</td>
<td>30.58</td>
<td>29.47</td>
<td>28.58</td>
</tr>
<tr>
<td>Lena</td>
<td>Noisy</td>
<td>24.92</td>
<td>20.41</td>
<td>18.55</td>
<td>15.27</td>
</tr>
<tr>
<td></td>
<td>DL</td>
<td>30.01</td>
<td>26.49</td>
<td>23.93</td>
<td>21.30</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>33.88</td>
<td>31.30</td>
<td>29.69</td>
<td>28.73</td>
</tr>
</tbody>
</table>

<p>| Table 2: Performance comparisons in terms of RMSE for the image texture part |
|-----------------------------|------------------------|------------------------|------------------------|------------------------|</p>
<table>
<thead>
<tr>
<th>Image</th>
<th>Method</th>
<th>$\sigma = 15$</th>
<th>$\sigma = 25$</th>
<th>$\sigma = 35$</th>
<th>$\sigma = 45$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barbara</td>
<td>DL</td>
<td>8.28</td>
<td>12.15</td>
<td>16.29</td>
<td>20.38</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>3.22</td>
<td>7.54</td>
<td>8.57</td>
<td>9.49</td>
</tr>
<tr>
<td>Lena</td>
<td>DL</td>
<td>8.08</td>
<td>12.14</td>
<td>16.09</td>
<td>20.60</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>5.16</td>
<td>6.94</td>
<td>8.36</td>
<td>9.33</td>
</tr>
</tbody>
</table>

<p>| Table 3: Performance comparisons in terms of MAE for the image texture part |
|-----------------------------|------------------------|------------------------|------------------------|------------------------|</p>
<table>
<thead>
<tr>
<th>Image</th>
<th>Method</th>
<th>$\sigma = 15$</th>
<th>$\sigma = 25$</th>
<th>$\sigma = 35$</th>
<th>$\sigma = 45$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barbara</td>
<td>DL</td>
<td>6.30</td>
<td>9.24</td>
<td>12.42</td>
<td>15.49</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>2.49</td>
<td>5.42</td>
<td>6.09</td>
<td>6.72</td>
</tr>
<tr>
<td>Lena</td>
<td>DL</td>
<td>6.09</td>
<td>9.23</td>
<td>12.21</td>
<td>15.60</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>3.49</td>
<td>4.65</td>
<td>5.50</td>
<td>6.12</td>
</tr>
</tbody>
</table>
3.2. Denoising of Image Structure

Table 4 shows the PSNR comparisons for the noise removal method of different image structure parts. The proposed method is better than the DL method on average about 4 dB and 2 dB in noise intensity $\sigma = 15$ and $\sigma = 45$, respectively. Therefore, the proposed method can improve the performance of the DL method for the image structure part in all noise corruption. Tables 5 and 6 show the RMSE and MAE comparisons of the noise removal methods for different image structures. The proposed method is only superior to the DL method in slight noise corruption, i.e., noise density $\sigma = 15$. But the proposed method cannot improve the DL method in high noise densities. This result is because the DL method uses the K-SVD algorithm which needs many iterations and is a time-consuming algorithm. Although the proposed method cannot improve the DL method in the image structure part in terms of RMSE, the overall performance of the proposed method is better than that of the DL method.

Fig. 6 shows the image structure. The structure part of a noisy image, which is corrupted by random noise with noise density $\sigma = 25$, is shown in Fig. 6 (b). By comparing the structure parts of the restored image shown in Figs. 6 (c) and (d), the image structure parts produced by the proposed method (Fig. 6 (d)) and the DL method (Fig. 6 (c)) are comparable. It should be noted that the texture produced by the proposed method is better than that of the DL method, in particular at the top-left corner of the images.

3.3. Restored Images

Fig. 7 shows the restored images of Barbara. This image is corrupted by random noise with noise density

![Fig. 6. Structure of the restored Barbara image: (a) noise-free image, (b) noisy image with noise density $\sigma = 25$, (c) structure of the denoised image using the DL method, and (d) structure of the denoised image using the proposed method.](image-url)
The DL and proposed methods can effectively reconstruct the noisy image. By comparing Figs. 7 (c) and (d), plenty of residual noise exists in the denoised image of the DL method. In addition, the denoised image shown in Fig. 7 (c) suffers from a blurred effect. On the contrary, the proposed method is better able to remove interference noise, while the details are well reconstructed. Therefore, the image quality denoised by the proposed method (Fig. 7 (d)) is much better than that denoised by the DL method. The major reason is due to the better restoration on the texture part of the image by using the proposed method.

![Fig. 7. Restored Barbara image: (a) noise-free image, (b) noisy image with noise density $\sigma = 25$, (c) denoised image using the DL method, and (d) denoised image using the proposed method.](image)

<table>
<thead>
<tr>
<th>Image</th>
<th>Method</th>
<th>$\sigma = 15$</th>
<th>$\sigma = 25$</th>
<th>$\sigma = 35$</th>
<th>$\sigma = 45$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barbara</td>
<td>Noisy</td>
<td>24.57</td>
<td>20.16</td>
<td>17.21</td>
<td>15.16</td>
</tr>
<tr>
<td></td>
<td>DL</td>
<td>29.61</td>
<td>26.72</td>
<td>24.19</td>
<td>21.34</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>32.25</td>
<td>30.51</td>
<td>27.03</td>
<td>24.98</td>
</tr>
<tr>
<td></td>
<td>Noisy</td>
<td>24.57</td>
<td>20.17</td>
<td>17.28</td>
<td>15.06</td>
</tr>
<tr>
<td>Lena</td>
<td>DL</td>
<td>29.76</td>
<td>26.64</td>
<td>23.92</td>
<td>21.17</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>32.65</td>
<td>30.27</td>
<td>27.20</td>
<td>24.92</td>
</tr>
</tbody>
</table>

Table 7: Performance comparisons in terms of PSNR for the restored images

4. Conclusions

This paper presented an effective approach to improving the DL method for random noise removal of an image. Initially improved MCA was employed to decompose the image into the image texture, structure, and edge parts. A BM3D method was used to restore the texture part of the image. In the image structure part, an ANLM algorithm was utilized. The image edge part was reconstructed by the K-SVD algorithm. Experimental results showed the proposed method could effectively remove background noise; meanwhile the image texture details were restored well.

References


---

**Der-Chang Tseng** received the B.S., M.S., and Ph.D. degrees in electronic engineering from National Taiwan University of Science and Technology, Taipei in 1989, 1995, and 2008, respectively. Currently, he is an associate professor with the Department of Electronic Engineering, National Taipei University of Technology, Taipei. His research interests are in the areas of secure communications and digital signal processing.
Ru-Yin Wei received her B.S. degree from the Department of Information and Telecommunications Engineering, Ming Chuan University, Taoyuan in 2015. She received her M.S. degree in electronic engineering from National Taipei University of Technology in 2017. Now she is a system engineer with Quanta Computer Inc., Taipei. Her research interests include image signal processing and multimedia applications.

Ching-Ta Lu received his B.S. and M.S. degrees in electronic engineering from National Taiwan University of Science and Technology in 1991 and 1995, respectively, and his Ph.D. degree in electrical engineering from National Tsing Hua University, Hsinchu in 2006. He had been with the Department of Electronic Engineering, Asia-Pacific Institute of Creativity, Miaoli from August 1995 to February 2008. He had been the Chair of the department in 2000 and 2006. He received the excellent teaching awards in 2006 and 2007, and the best tutor awards in 1986, 1987, 2005, 2006, and 2007 from Asia Pacific Institute of Creativity. He also received the excellent teaching awards in 2015 and 2016, and the best tutor awards in 2009, 2013, and 2015 from Asia University, Taichung. He has been a full professor with the Department of Information Communication, Asia University since February 2008. He has published more than 40 journal papers and 70 conference papers. His current research interests include speech enhancement, image denoising, speech coding, speech and image signal processing, and the applications of artificial intelligence.

Ling-Ling Wang received the B.S., M.S., and Ph.D. degrees in computer science and information engineering from National Chiao Tung University, Hsinchu in 1984, 1986, and 1990, respectively. From 1986 to 1987, she was an associate engineer with the System Software Department, ERSO/ITRI, Hsinchu. From 1991 to 1997, she was an associate professor in computer science with National Tsing Hua University. Currently, she is a professor in information communications with Asia University. Her major research is in image processing and artificial intelligence.