Denoising performance analysis of adaptive decision based inverse distance weighted interpolation (DBIDWI) algorithm for salt and pepper noise

Vorapoj Patanavijit
Assumption University, Thailand

ABSTRACT
Due to its superior performance for denoising an image, which is contaminated by impulsive noise, an adaptive decision based inverse distance weighted interpolation (DBIDWI) algorithm is one of the most dominant and successful denoising algorithm, which is recently proposed in 2017, however this DBIDWI algorithm is not desired for denoising the full dynamic intensity range image, which is comprised of min or max intensity. Consequently, the research article aims to study the performance and its limitation of the DBIDWI algorithm when the DBIDWI algorithm is performed in both general images and the images, which are comprised of min or max intensity. In this simulation experiments, six noisy images (Lena, Mobile, Pepper, Pentagon, Girl and Resolution) under salt&pepper noise are used to evaluate the performance and its limitation of the DBIDWI algorithm in denoised image quality (PSNR) perspective.

Keywords: Digital image denoising, DBIDWI (decision based inverse distance weighted interpolation), SMF (standard median filtering)

1. GENERAL OVERVIEW
In general, an impulsive noise is created in a digital image [1-3] because of camera sensor malfunction or communication fault therefore many denoising algorithms [4-24] have been invented for advance applications [25-28]. One of the most dominant and successful denoising algorithms is the standard median filter (SMF) [4-6], which is invented for denoising salt and pepper noise however its performance is limited because the SMF is processed all pixels (both noisy and noiseless). Later, the alternative denoising algorithms [7-24] based on detecting and denoising techniques are intensively invented for improving denoising performance. Recently, one of the most powerful and effective denoising algorithm is an adaptive decision based inverse distance weighted interpolation (DBIDWI) algorithm [29], which is proposed in 2017. Due to the constrain of its characteristic process, this denoising algorithm can only be performed on image, which is comprised of min or max intensity range thereby the research article aims to study the performance and its limitation of the DBIDWI algorithm.

The research article is aligned as follow: the general overview is offered in section 1 and the main concept of DBIDWI (Decision Based Inverse Distance Weighted Interpolation) is offered in section 2. Later, the comprehensive simulated consequence and its experimental outline are offered in section 3 and section 4, respectively.

Journal homepage: http://iaescore.com/journals/index.php/ijeecs
2. **THE MAIN CONCEPT OF DBIDWI (DECISION BASED INVERSE DISTANCE WEIGHTED INTERPOLATION)**

The denoising algorithm based on DBIDWI algorithm [29] compounds of the detecting and denoising technique as offering in the following sub-section. Overall flowchart of denoising sub-process based on DBIDWI (Decision Based Inverse Distance Weighted Interpolation) as shown in Figure 1.

![Overall flowchart of denoising sub-process based on DBIDWI](image)

Figure 1. Overall flowchart of denoising sub-process based on DBIDWI (Decision Based Inverse Distance Weighted Interpolation)
2.1. Detecting Sub-Process of the DBIDWI Based Denoising Algorithm

At first, the detecting process of the DBIDWI based denoising algorithm simply checks every pixel and defined that pixel is noisy \( \text{NDM}(i, j) = 1 \) if the pixel intensity is min (0) or max (255) in dynamic range otherwise the pixel is noiseless \( \text{NDM}(i, j) = 0 \).

2.2. Denoising Sub-Process of the DBIDWI Based Denoising Algorithm

Step 1. The denoising sub-process filters only noisy pixels, which are classified from the previous detecting sub-process, by creating the calculated window \( W_{y,\text{win}} \), which is initially set at 3x3 (or \( n = 3 \)) with center at noisy pixel \( y(i, j) \) and, later, the noiseless pixels are counted in that window \( W_{y,\text{win}} \). Support that noiseless pixels \( N_{\text{noiseless, pixels}} \) are counted and less than 3 pixels (in order to prevent blur and unreliable case) therefore calculated window expands by 1 pixel (as shown in following figure) and the noiseless pixels are recounted in the expanded window \( W_{y,\text{win}} \) again until there are more than 3 noiseless pixels in the expanded window.

Step 2. Support that there are more than 3 noiseless pixels in the expanded window therefore the inverse distance of noiseless pixels \( d(n_{\text{noiseless, pixels}}) \) is computed as following equation:

\[
d(n_{\text{noiseless, pixels}}) = d_n(n_{\text{noiseless, pixels}}) = \left( \frac{1}{d(i-j_{\text{noiseless, pixels}})} \right) + \left( \frac{1}{d(i-j_{\text{noiseless, pixels}})} \right) \quad \text{for} \quad n_{\text{noiseless, pixels}} = 1, 2, \ldots, N_{\text{noiseless, pixels}}
\]

Step 3. The noisy pixel \( y(i, j) \) are replacing denoised pixel \( \hat{x}(i, j) \), which is computed as following equation:

\[
\hat{x}(i, j) = \sum_{n_{\text{noiseless, pixels}}=1}^{N_{\text{noiseless, pixels}}} d_n(n_{\text{noiseless, pixels}}) \times W_y(n_{\text{noiseless, pixels}})
\]

Where,

\[
d_n(n_{\text{noiseless, pixels}}) = d(n_{\text{noiseless, pixels}}) = \sum_{i=1}^{N_{\text{noiseless, pixels}}} d(n_{\text{noiseless, pixels}})
\]

The overall flowchart of denosing sub-process based on DBIDWI (Decision Based Inverse Distance Weighted Interpolation) can be appeared as following figure.

3. COMPUTATIONAL EXAMPLES
3.1. EXAMPLE 1

Support that the the calculated window \( W_{y,\text{win}} \) of the interested noisy pixel \( y(i, j) \) can be formulated as following.

| \( y(i-k, j-l) \) | \( y(i, j-1) \) | \( y(i+k, j-l) \) |
|----------------|----------------|----------------|
| 125            | 131            | 118            |
| \( y(i-1, j) \) | \( y(i, j) \)  | \( y(i+1, j) \) |
| 0              | 255            | 0              |
| \( y(i-k, j+l) \) | \( y(i, j+1) \) | \( y(i+1, j+l) \) |
| 120            | 0              | 255            |
and the noise detected matrix of the calculated window can be formulated as following.

\[
\text{NDM} = \begin{bmatrix}
0 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 1
\end{bmatrix}
\]

**Step 1.** The denoising sub-process filters only noisy pixels, which are classified from the previous detecting sub-process, by creating the calculated window.

From noise detected matrix can **NDM**, the noiseless pixels are counted in that window \( \text{NDM} \), therefore \( \text{N}_{\text{noiseless\_pixels}} = 3 \).

**Step 2.** Support that there are more than 3 noiseless pixels in the expanded window therefore the inverse distance of noiseless pixels \( d\left(\text{n}_{\text{noiseless\_pixels}}\right) \) is computed as following equation:

\[
d\left(\text{n}_{\text{noiseless\_pixels}}\right) = \left(\sum_{i=1}^{n_{\text{noiseless\_pixels}}} d_{n_{\text{noiseless\_pixels}}} \right)_{i=1,2,\ldots,n_{\text{noiseless\_pixels}}}
\]

\[
d = \begin{bmatrix}
0.5359 & 1 & 0.5359 \\
1 & 0 & 1 \\
0.5359 & 1 & 0.5359
\end{bmatrix}
\Rightarrow
d\left(\text{n}_{\text{noiseless\_pixels}}\right) = \begin{bmatrix}
0.5359 & 0 & 0.5359 \\
0 & 0 & 0 \\
0.5359 & 0 & 0
\end{bmatrix}
\]

Therefore,

\[
d_{N}(\text{n}_{\text{noiseless\_pixels}}) \times \text{W}_{Y}(\text{n}_{\text{noiseless\_pixels}}) = \begin{bmatrix}
125 & 131 & 118 \\
0 & 255 & 0 \\
120 & 0 & 255
\end{bmatrix} \times \begin{bmatrix}
0.5359 & 0.5359 \\
0 & 0.5359 \\
0.5359 & 0
\end{bmatrix} = \begin{bmatrix}
41.66 & 0 & 39.33 \\
0 & 0 & 0 \\
40 & 0 & 0
\end{bmatrix}
\]

**Step 3.** The noisy pixel \( y(i,j) \) are replaced by the denoised pixel \( \hat{x}(i,j) \), which is computed as following equation:

\[
\hat{x}(i,j) = \sum_{n_{\text{noiseless\_pixels}}} d_{N}(\text{n}_{\text{noiseless\_pixels}}) \times \text{W}_{Y}(\text{n}_{\text{noiseless\_pixels}})
\]

\[
\hat{x}(i,j) = \left(41.66 + 39.33 + 40\right) = 121
\]

3.2. **EXAMPLE 2**

Support that the the calculated window \( \text{W}_{Y\_x\_y} \) of the interested noisy pixel \( y(i,j) \) can be formulated as following.
and the noise detected matrix of the calculated window can be formulated as following.

\[
\text{NDM} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 \\
\end{bmatrix}
\]

**Step 1.** The denosing sub-process filters only noisy pixels, which are classified from the previous detecting sub-process, by creating the calculated window.

\[
W_{Y,3,3} = \begin{bmatrix}
255 & 255 & 0 \\
0 & 255 & 255 \\
112 & 114 & 255 \\
\end{bmatrix}
\quad \text{and} \quad
\text{NDM} = \begin{bmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
0 & 0 & 1 \\
\end{bmatrix}
\]

From noise detected matrix can \(\text{NDM}\), the noiseless pixels are counted in that window \(W_{Y,3,3}\) therefore \(N_{\text{noiseless, pixels}} = 2\).

Support that noiseless pixels \(N_{\text{noiseless, pixels}} = 2\) are counted and less than 3 pixels therefore calculated window expands by 1 pixel.

\[
W_{Y,3,5} = \begin{bmatrix}
0 & 0 & 255 & 0 & 255 \\
0 & 255 & 255 & 0 & 118 \\
0 & 0 & 255 & 255 & 0 \\
255 & 112 & 255 & 255 & 255 \\
255 & 0 & 0 & 255 & 111 \\
\end{bmatrix}
\quad \text{and} \quad
\text{NDM} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 \\
\end{bmatrix}
\]

From noise detected matrix can \(\text{NDM}\), the noiseless pixels are counted in that window \(W_{Y,3,5}\) therefore \(N_{\text{noiseless, pixels}} = 4\).

**Step 2.** Support that there are more than 3 noiseless pixels in the expanded window therefore the inverse distance of noiseless pixels \(d\left(N_{\text{noiseless, pixels}}\right)\) is computed as following equation:
Denoising performance analysis of adaptive decision based inverse distance... (Vorapoj Patanavijit)

\[
d(d_{\text{noiseless}}) = \left( \sum_{i_{\text{noiseless}} \text{ pixel}} + \sum_{j_{\text{noiseless}} \text{ pixel}} \right)^{d_b} \]

\[
d = \begin{bmatrix}
0.1539 & 0.2349 & 0.2872 & 0.2349 & 0.1539 \\
0.2349 & 0.5359 & 1 & 0.5359 & 0.2349 \\
0.2872 & 1 & 0 & 1 & 0.2872 \\
0.1539 & 0.2349 & 0.2872 & 0.2349 & 0.1539
\end{bmatrix}
\]

Therefore,

\[
d(d_{\text{noiseless}}) \times W_y(d_{\text{noiseless}}) = \begin{bmatrix}
0 & 0 & 255 & 0 & 255 \\
0 & 255 & 255 & 0 & 118 \\
0 & 0 & 255 & 255 & 0 \\
255 & 112 & 114 & 255 & 255 \\
255 & 0 & 0 & 255 & 111
\end{bmatrix}
\]

\[
d(d_{\text{noiseless}}) \times W_y(d_{\text{noiseless}}) = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 14.40 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 59.23 & 0 & 0 \\
0 & 0 & 0 & 8.87 & 0
\end{bmatrix}
\]

**Step 3.** The noisy pixel \( y(i, j) \) are replaced the denoised pixel \( \hat{x}(i, j) \), which is computed as following equation:

\[
\hat{x}(i, j) = \sum_{n_{\text{noiseless}} \text{ pixel}} d_y(n_{\text{noiseless}} \text{ pixel}) \times W_y(n_{\text{noiseless}} \text{ pixel}) \\
\hat{x}(i, j) = (31.18 + 59.23 + 14.40 + 8.87) = 113.69
\]

4. **COMPREHENSIVE SIMULATED CONSEQUENCE**

The numerical experiment is conducted by using MATLAB program on six simulated data, which are comprised of Lena (256x256), Pepper (256x256), Resolution (128x128), Girl-Tiffany (256x256), Baboon (256x256), House (128x128), used to evaluate the upper and lower range of DBIDWI performance. First, all original data are added by Salt and Pepper Noise from 5% to 90% for forming the noisy data. Later these noisy data are filtered to suppress Salt and Pepper Noise by DBIDWI algorithm. From the numerical consequences, the quality measurement (PSNR) of the denoised image by DBIDWI algorithm are indicated in Table 1 for Lena (256x256), Pepper (256x256), Resolution (128x128) and Table 2 for Girl-Tiffany (256x256), Baboon (256x256), House (128x128). The DBIDWI algorithms can improve the image quality in almost all simulated data, except for Resolution (128x128) because the Resolution image is comprised of max ("255") and min ("0") in intensity dynamic range.

5. **CONCLUSION**

This research article aims to exhaustively evaluate the upper and lower range of DBIDWI performance, one of the most dominant and successful denoising algorithm, which is recently proposed in 2017, under Salt and Pepper Noise at several density. Comprehensive simulated consequences conduct on six simulated data, which are comprised of Lena (256x256), Pepper (256x256), Resolution (128x128), Girl-Tiffany (256x256), Baboon (256x256), House (128x128). Due to the limitation of noise detection process of the DBIDWI algorithms, the DBIDWI algorithms has obviously improve the image quality...
(PSNR) in almost all simulated data, except for Resolution (128x128) because the Resolution image is comprised of max and min in intensity dynamic range.

ACKNOWLEDGEMENTS

The research project was funded by Assumption University. Comprehensive Simulated Consequence of Salt and Pepper Noise (Lena, Pepper, Resolution) as shown in Table 1 and Comprehensive Simulated Consequence of Salt and Pepper Noise (Girl, Babool, House) as shown in Table 2.

Table 1. Comprehensive Simulated Consequence of Salt and Pepper Noise (Lena, Pepper, Resolution)

| Tested Images | SPN | Noise Density | Observed Image | Denoising Algorithm | SMF | DBIDWI |
|---------------|-----|---------------|----------------|---------------------|-----|--------|
| Lena (256x256) | 5   | 18.7139       | 31.6421        | 43.7281             |     |        |
|                | 10  | 15.6564       | 30.7076        | 40.5308             |     |        |
|                | 15  | 13.8274       | 29.2982        | 38.6418             |     |        |
|                | 20  | 12.6389       | 27.6257        | 37.2258             |     |        |
|                | 25  | 11.6783       | 25.4101        | 35.9382             |     |        |
|                | 30  | 10.8971       | 23.6811        | 35.1593             |     |        |
|                | 35  | 10.2240       | 20.8127        | 34.1636             |     |        |
|                | 40  | 9.6481        | 19.0080        | 33.3373             |     |        |
| Pepper (256x256) | 45  | 9.0745        | 16.8389        | 32.7176             |     |        |
|                | 50  | 8.6553        | 15.4758        | 32.1273             |     |        |
|                | 55  | 8.2118        | 13.8573        | 31.4207             |     |        |
|                | 60  | 7.7813        | 12.3280        | 30.8325             |     |        |
|                | 65  | 7.4884        | 11.3251        | 30.1591             |     |        |
|                | 70  | 7.1697        | 10.2861        | 29.5147             |     |        |
|                | 75  | 6.8497        | 9.1271         | 28.7243             |     |        |
|                | 80  | 6.5846        | 8.3311         | 27.9712             |     |        |
|                | 85  | 6.3241        | 7.5344         | 27.3899             |     |        |
|                | 90  | 6.0604        | 6.8241         | 26.1503             |     |        |
|                | 5   | 18.4752       | 32.2578        | 45.2269             |     |        |
|                | 10  | 15.3798       | 30.6116        | 42.3736             |     |        |
|                | 15  | 13.5570       | 28.8470        | 40.2444             |     |        |
|                | 20  | 12.3593       | 26.5888        | 38.9573             |     |        |
|                | 25  | 11.3929       | 24.2073        | 37.7392             |     |        |
|                | 30  | 10.6242       | 22.0663        | 36.7617             |     |        |
|                | 35  | 9.9742        | 20.3774        | 36.0150             |     |        |
|                | 40  | 9.3998        | 18.4321        | 35.0674             |     |        |
| Pepper (128x128) | 45  | 8.8599        | 16.6168        | 34.1463             |     |        |
|                | 50  | 8.3843        | 14.8506        | 33.3663             |     |        |
|                | 55  | 7.9930        | 13.4655        | 32.8686             |     |        |
|                | 60  | 7.6189        | 12.0128        | 32.1738             |     |        |
|                | 65  | 7.2684        | 10.8920        | 31.5987             |     |        |
|                | 70  | 6.9246        | 9.7704         | 30.7429             |     |        |
|                | 75  | 6.6418        | 8.8751         | 29.8355             |     |        |
|                | 80  | 6.3710        | 8.0166         | 29.0865             |     |        |
|                | 85  | 6.1097        | 7.2402         | 28.0305             |     |        |
|                | 90  | 5.8582        | 6.5767         | 27.0502             |     |        |
|                | 5   | 16.1344       | 18.2861        | 8.6930              |     |        |
|                | 10  | 13.4819       | 17.9425        | 8.5201              |     |        |
|                | 15  | 11.4968       | 17.0880        | 8.4935              |     |        |
|                | 20  | 10.1271       | 16.2124        | 8.3813              |     |        |
|                | 25  | 9.2699        | 15.2214        | 8.2850              |     |        |
|                | 30  | 8.4430        | 14.4548        | 8.1603              |     |        |
|                | 35  | 7.9307        | 13.6304        | 8.2997              |     |        |
|                | 40  | 7.3308        | 12.6223        | 7.7616              |     |        |
| Resolution (128x128) | 45  | 6.6368        | 11.3597        | 8.0067              |     |        |
|                | 50  | 6.2938        | 10.4851        | 7.9123              |     |        |
|                | 55  | 5.8134        | 9.4501         | 7.8132              |     |        |
|                | 60  | 5.4436        | 8.5925         | 7.3368              |     |        |
|                | 65  | 5.0707        | 7.6495         | 7.4990              |     |        |
|                | 70  | 4.6795        | 6.6295         | 7.6584              |     |        |
|                | 75  | 4.5178        | 6.3093         | 7.2709              |     |        |
|                | 80  | 4.1940        | 5.3585         | 7.1479              |     |        |
|                | 85  | 3.9342        | 4.7261         | 7.1875              |     |        |
|                | 90  | 3.7113        | 4.2234         | 6.7209              |     |        |
Table 2. Comprehensive Simulated Consequence of Salt and Pepper Noise (Girl, Babool, House)

| Tested Images | SPN | PSNR (dB) |
|---------------|-----|-----------|
|               | Noise Density | Observed Image | Denoising Algorithm | SMF | DBIDWI |
| Girl (256x256) | 5   | 16.4490 | 32.4667 | 39.3666 |
|               | 10  | 13.6890 | 31.5583 | 38.4352 |
|               | 15  | 11.9287 | 27.6179 | 37.2170 |
|               | 20  | 10.6567 | 25.5153 | 36.5714 |
|               | 25  | 9.5498  | 22.9614 | 35.6831 |
|               | 30  | 8.8677  | 20.7738 | 35.3124 |
|               | 35  | 8.0984  | 18.4410 | 34.2244 |
|               | 40  | 7.5978  | 16.5146 | 33.9797 |
|               | 45  | 7.0728  | 14.8145 | 33.5408 |
|               | 50  | 6.5712  | 13.0319 | 32.5247 |
|               | 55  | 6.2085  | 11.8226 | 32.2016 |
|               | 60  | 5.8609  | 10.4981 | 31.7084 |
|               | 65  | 5.4832  | 9.1396  | 30.9636 |
|               | 70  | 5.1311  | 8.0463  | 30.4654 |
|               | 75  | 4.8712  | 7.1994  | 29.8662 |
|               | 80  | 4.5674  | 6.2520  | 29.0774 |
|               | 85  | 4.3054  | 5.4218  | 28.0161 |
|               | 90  | 4.0573  | 4.7465  | 27.2722 |

| Baboon (256x256) | 45  | 7.0728 | 14.8145 | 33.5408 |
|                  | 50  | 6.5712 | 13.0319 | 32.5247 |
|                  | 55  | 6.2085 | 11.8226 | 32.2016 |
|                  | 60  | 5.8609 | 10.4981 | 31.7084 |
|                  | 65  | 5.4832 | 9.1396  | 30.9636 |
|                  | 70  | 5.1311 | 8.0463  | 30.4654 |
|                  | 75  | 4.8712 | 7.1994  | 29.8662 |
|                  | 80  | 4.5674 | 6.2520  | 29.0774 |
|                  | 85  | 4.3054 | 5.4218  | 28.0161 |
|                  | 90  | 4.0573 | 4.7465  | 27.2722 |

| House (128x128) | 45  | 7.0728 | 14.8145 | 33.5408 |
|                 | 50  | 6.5712 | 13.0319 | 32.5247 |
|                 | 55  | 6.2085 | 11.8226 | 32.2016 |
|                 | 60  | 5.8609 | 10.4981 | 31.7084 |
|                 | 65  | 5.4832 | 9.1396  | 30.9636 |
|                 | 70  | 5.1311 | 8.0463  | 30.4654 |
|                 | 75  | 4.8712 | 7.1994  | 29.8662 |
|                 | 80  | 4.5674 | 6.2520  | 29.0774 |
|                 | 85  | 4.3054 | 5.4218  | 28.0161 |
|                 | 90  | 4.0573 | 4.7465  | 27.2722 |

REFERENCES

[1] J. Astola and P. Kuosmanen, “Fundamentals of Nonlinear Digital Filtering”, CRC Press, Boca Raton, Fla, USA, 1997.
[2] R. C. Gonzalez and R. E. Woods, “Digital Image Processing”, Prentice-Hall, Upper Saddle River, NJ, USA, 2nd edition, 2002.
[3] M. H. Hayes, “Statistical Digital Signal Processing and Modeling”, JohnWiley & Sons, Singapore, 2002.
[4] W. K. Pratt, “Median filtering,” Tech. Rep., Image Proc. Inst., Univ. Southern California, Los Angeles, Sep. 1975.
[5] N. C. Gallagher Jr. and G. L. Wise, “A Theoretical Analysis of the Properties of Median Filters,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 29, no. 6, pp. 1136-1141, 1981.
[6] T. A. Nodes and N. C. Gallagher Jr., “Median Filters: Some Modifications and Their Properties,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 30, no. 5, pp. 739-746, 1982.

Denoising performance analysis of adaptive decision based inverse distance... (Vorapoj Patanavijit)
[7] E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, “A New Efficient Approach for the Removal of Impulse Noise from Highly Corrupted Images,” IEEE Transactions on Image Processing, vol. 5, no. 6, pp. 1012-1025, 1996.

[8] D. R. K. Brownrigg, “The weighted median filter,” Communications of the ACM, vol. 27, no. 8, pp. 807-818, 1984.

[9] O. Yli-Harja, J. Astola, and Y. Neuvo, “Analysis of the Properties of Median and Weighted Median Filters Using Threshold Logic and Stack Filter Representation,” IEEE Transactions on Signal Processing, vol. 39, no. 2, pp. 395-410, 1991.

[10] G. R. Arec and J. L. Paredes, “Recursive Weighted Median Filters Admitting Negative Weights and Their Optimization,” IEEE Transactions on Signal Processing, vol. 48, no. 3, pp. 768-779, 2000.

[11] Y. Dong and S. Xu, “A New Directional Weighted Median Filter for Removal of Random-Valued Impulse Noise,” IEEE Signal Processing Letters, vol. 14, no. 3, pp. 193-196, 2007.

[12] T. Chen, K.-K. Ma, and L.-H. Chen, “Tri-State Median Filter for Image Denoising,” IEEE Transactions on Image Processing, vol. 8, no. 12, pp. 1834-1838, 1999.

[13] H. Hwang and R. A. Haddad, “Adaptive Median Filters: New Algorithms and Results,” IEEE Transactions on Image Processing, vol. 4, no. 4, pp. 499-502, 1995.

[14] S. Zhang and M. A. Karim, “A New Impulse Detector for Switching Median Filters,” IEEE Signal Processing Letters, vol. 9, no. 11, pp. 360-363, 2002.

[15] H.-L. Eng and K.-K. Ma, “Noise Adaptive Soft-Switching Median Filter,” IEEE Transactions on Image Processing, vol. 10, no. 2, pp. 242-251, 2001.

[16] Z. Wang and D. Zhang, “Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images,” IEEE Transactions on Circuits and Systems II, vol. 46, no. 1, pp. 78-80, 1999.

[17] P.-E. Ng and K.-K. Ma, “A Switching Median Filter with Boundary Discriminative Noise Detection for Extremely Corrupted Images,” IEEE Transactions on Image Processing, vol. 15, no. 6, pp. 1506-1516, 2006.

[18] R. H. Chan, C.-W. Ho, and M. Nikolova, “Salt-and-Pepper Noise Removal by Median-Type Noise Detectors and Detailpreserving Regularization,” IEEE Transactions on Image Processing, vol. 14, no. 10, pp. 1479-1485, 2005.

[19] K. S. Srinivasan and D. Ebenezer, “A New Fast and Efficient Decision-Based Algorithm for Removal of High-Density Impulse Noises,” IEEE Signal Processing Letters, vol. 14, no. 3, pp. 189-192, 2007.

[20] S. Schulze, M. Nachtegael, V. DeWitte, D. van der Weken, and E. E. Kerre, “A Fuzzy Impulse Noise Detection and Reduction Method,” IEEE Transactions on Image Processing, vol. 15, no. 5, pp. 1153-1162, 2006.

[21] A. Ben Hamza and H. Krim, “Image Denoising: A Nonlinear Robust Statistical Approach,” IEEE Transactions on Signal Processing, vol. 49, no. 12, pp. 3045-3054, 2001.

[22] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image Quality Assessment: from Error Visibility to Structural Similarity,” IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, 2004.

[23] E. Pavankumar, Manoj Kumar Rajgopal, Robust Visual Multi-Target Trackers: A Review, Indonesian Journal of Electrical Engineering and Computer Science(IJECS), Vol. 12, No. 1, October 2018, pp. 7-16.

[24] Nur Ateqa Binti Mat Kasim, Nur Hidayah Binti Abd Rahman, Zaidah Ibrahim, Nur Nabilah Abu Mangshor, Celebrity Face Recognition using Deep Learning, Indonesian Journal of Electrical Engineering and Computer Science(IJECS), vol. 12, No. 2, November 2018, pp. 476-481.

[25] Raden Arie Setiyawan, Rudy Sunoko, Mohdhammad Agus Choiron, Panca Mudji Rahardjo, “Implementation of Stereo Vision Semi-Global Block Matching Methods for Distance Measurement”, Indonesian Journal of Electrical Engineering and Computer Science(IJECS), Vol. 12, No. 2, November 2018, pp. 585-591.

[26] Vorapoj Palanavijit, “Performance Analysis of Denoising Algorithm Based on Adaptive Median Filter Under Unsystematic Intensity Impulse and Salt&Pepper Noise”, The 6th International Electrical Engineering Congress (iEECON2017), Krabi, Thailand, March 2018.

[27] V. Jayaraj and D. Ebenezer, “A New Switching-BasedMedian Filtering Scheme and Algorithm for Removal of High-Density Salt and Pepper Noise in Images”, EURASIP Journal on Advances in Signal Processing, Hindawi Publishing Corporation, 2010.

[28] P. Zhang, F. Li, “A New Adaptive Weighted Mean Filter for Removing Salt-Andpepper Noise”. IEEE Signal Process Lett. 21(10), 1280–1283 (2014).

[29] V. Kishorebabu et al, “An Adaptive Decision Based Interpolation Scheme for the Removal of High Density Salt and Pepper Noise in Images”, EURASIP Journal on Image and Video Processing, 2017, pp. 1-18.
Vorapoj Patanavijit received the B.Eng., M.Eng. and Ph.D. degrees from the Department of Electrical Engineering at the Chulalongkorn University, Bangkok, Thailand, in 1994, 1997 and 2007 respectively. He has served as a full-time lecturer at Department of Electrical and Electronic Engineering, Faculty of Engineering, Assumption University since 1998 where he is currently an Associate Professor. He has authored and co-authored over 150 national/international peer-reviewed publications in Digital Signal Processing (DSP) and Digital Image Processing (DIP). He received the best paper awards from many conferences such as ISCTIT2006, NCTIT2008, EECON-33 (2010), EECON-34 (2011), EECON-35 (2012) and etc. Moreover, he is invited to be the guest speaker at IWAIT2014 and contributed the invited paper at iECON 2014. He has served as a Technical Program Committees (TPC) on Signal Processing of ECTI (Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology) Association, Thailand since 2012 to 2015. As a technical reviewer of international journals since 2006, he has been assigned to review over 60 journal papers (indexed by ISI) from IEEE Transactions on Image Processing, IEEE Journal of Selected Topics in Signal Processing (J-STSP), IET in Image Processing (IEEE), IEEE Signal Processing Letters (IEEE), EURASIP Journal on Applied Signal Processing (JASP), Digital Signal Processing (Elsevier Ltd.), Journals of Neurocomputing (Elsevier Ltd.), Neural Networks (Elsevier Ltd.), International Journal for Light and Electron Optics (Optik) (Elsevier Ltd.), The Visual Computer (Springer), Journal of Electronic Imaging (SPIE), Journal of Optical Engineering (SPIE), IEICE Journal Electronics Express (ELEX) and ECTI Transactions on CIT (ECTI Thailand). As a technical reviewer of over 40 international/national conferences since 2006, he has been assigned to review over 130 proceeding papers. He has participated in more than 8 projects and research programmed funded by public and private organizations. He works in the field of signal processing and multidimensional signal processing, specializing, in particular, on Image/Video Reconstruction, SRR (Super-Resolution Reconstruction), Compressive Sensing, Enhancement, Fusion, Digital Filtering, Denoising, Inverse Problems, Motion Estimation, Optical Flow Estimation and Registration.