CHAPTER 2

REMOVAL OF HIGH DENSITY IMPULSE NOISE USING MORPHOLOGICAL BASED ADAPTIVE UNSYMMETRICAL TRIMMED MID-POINT FILTER

2.1 INTRODUCTION

Digital images are often corrupted by impulse noise during acquisition, storage and transmission. The impulse noise can be classified under: salt and pepper noise and random valued noise. The pixel which takes either maximum or minimum intensity value (0 or 255) is classified as salt and pepper noise and if the pixel takes any value between 0 and 255, it is classified as random valued impulse noise. Though there are various algorithms for the removal of salt and pepper noise, they are not efficient at higher noise densities, hence effort is taken to design an efficient algorithm for removal of salt and pepper noise in images particularly at high noise densities.

A Modified Decision Based Unsymmetric Trimmed Median Filter (MDBUTMF) algorithm for removal of salt and pepper noises in images corrupted with high noise densities was proposed by Esakkirajan et al (2011). The main challenge is that if all the pixels in processing window are corrupted, the algorithm replaces the centre pixel with the mean of the processing window which blurs the image at high noise densities.

A Morphological Based Adaptive Unsymmetrical Trimmed Mid-Point Filter (MAUTMPF) is proposed to overcome this drawback, in which the corrupted pixels in a selected window are trimmed and the mid-point of the
minimum and maximum pixel value of remaining uncorrupted pixels is selected with the essential condition that atleast two pixels in the selected window are uncorrupted. If not, the size of the window is increased and the iterations are stopped when the size of the window reaches to 7.

2.2 PROPOSED MAUTMPF

The MDBUTMF replaces the noisy pixels in a selected window by taking either mean (Mean Filter) or median (Median Filter) of the remaining pixels after trimming 0 and 255. However, the mean filter blurs the image both at low and high noise densities and median filter doesn’t preserve edges for higher Noise Density (ND). In this chapter, a MAUTMPF is proposed to overcome the above drawbacks. In the proposed MAUTMPF, 3 X 3 window is selected and the corrupted pixels are trimmed unsymmetrically. If the remaining pixels in the selected window are greater than or equal to 2, then the midpoint of the maximum and minimum value is taken and replaced by the centre pixel. On the other hand, if the remaining pixels in the selected window is less than 2, then the window size is increased by 2 and the same procedure is repeated. The iterations in the algorithm are continued till window size reaches 7 X 7. If an estimate for noisy pixel can’t be reached in a 7 X 7 window also, then the centre pixel is replaced by mid-point of minimum and maximum intensity values of already processed pixels in the initial selected 3 x 3 window.

2.2.1 MAUTMPF Algorithm

The steps involved in the MAUTMPF algorithm are given below.

Steps

1. Consider the noisy image.
2. Morphological operators ‘erosion’ and ‘dilation’ are used to find the minimum ($Z_{min}=0$) and maximum ($Z_{max}=255$) values respectively, which are assumed to be corrupted pixels.
3. Consider a 3 x 3 window with $P_{ij}$ be the center pixel.
4. If $P_{ij}$ is uncorrupted, slide the window to the next pixel. If corrupted,
5. Check whether the processing window contains at least two uncorrupted pixels, If yes, go to step 8.
6. Increase the window size by 2 till the window size reaches 7 x 7 and follow step 5.
7. If all the pixels in 7 x 7 are corrupted, go to step 9.
8. Replace $P_{ij}$ by the mid-point value of minimum and maximum values in the processing window.
9. $P_{ij}$ is replaced by the mid-point value of minimum and maximum values of already processed pixels in the 3 x 3 window around $P_{ij}$.

In step 2, two morphological operators are used viz., erosion and dilation to detect the minimum and maximum pixel values (0 and 255) which are considered to be corrupted pixels.

The flowchart representation of detailed working of the MAUTMPF is given in Figure 2.1.
2.2.2 Illustration of MAUTMPF Algorithm

The processing of the algorithm starts from the first pixel and moves pixel by pixel till all the pixels in the image are covered. The iterations of the MAUTMPF algorithm are discussed in different cases.
Case i) If the processing pixel (represented as Pₙ) is noisy and the number of uncorrupted pixels in selected 3 X 3 window is greater than 2.

\[
\begin{bmatrix}
0 & 99 & 110 \\
0 & 0 & 104 \\
255 & 255 & 255
\end{bmatrix}
\]

Min. value:99; Max value:110
Midpoint value = (99+110)/2 = 104.5
Pₙ is replaced by 104.5

\[
\begin{bmatrix}
0 & 99 & 110 \\
0 & 104.5 & 104 \\
255 & 255 & 255
\end{bmatrix}
\]

Actual image value is 104

Case ii) If the processing pixel is noisy and the number of uncorrupted pixels in selected 3 X 3 window is less than 2, window size is increased by 2.

\[
\begin{bmatrix}
255 & 0 & 104 \\
255 & 0 & 0 \\
255 & 0 & 255
\end{bmatrix}
\]

Only one pixel is uncorrupted, so the window size is incremented by 2, i.e., consider 5 X 5 window.

\[
\begin{bmatrix}
255 & 255 & 0 & 0 & 255 \\
255 & 255 & 0 & 104 & 97 \\
0 & 255 & 0 & 0 & 0 \\
0 & 255 & 0 & 255 & 110 \\
97 & 0 & 0 & 0 & 255
\end{bmatrix}
\]

Min value:97; Max value:110
Midpoint value = (97+110)/2 = 103.5
Pₙ replaced by 103.5
Case (iii) $P_j$ is corrupted and less than 2 uncorrupted pixels in the selected 5 X 5 window

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>255</td>
<td>255</td>
<td>0</td>
<td>0</td>
<td>255</td>
<td></td>
</tr>
<tr>
<td>255</td>
<td>255</td>
<td>0</td>
<td>104</td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>255</td>
<td>103.5</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>255</td>
<td>0</td>
<td>255</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>255</td>
<td></td>
</tr>
</tbody>
</table>

Actual image value is 104

Only one pixel is uncorrupted, so the window size is incremented by 2. i.e., consider 7 X 7

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>255</td>
<td>255</td>
<td>255</td>
<td>0</td>
<td>255</td>
<td>0</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>255</td>
<td>0</td>
<td>255</td>
<td>0</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>255</td>
<td>115</td>
<td>255</td>
<td>255</td>
<td>0</td>
<td>0</td>
<td>255</td>
<td>255</td>
</tr>
<tr>
<td>0</td>
<td>255</td>
<td>255</td>
<td>255</td>
<td>0</td>
<td>0</td>
<td>255</td>
<td>255</td>
</tr>
</tbody>
</table>

Min value :204 ; Max value :219

Midpoint value = (204+219)/2 = 211.5

$P_j$ replaced by 211.5

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>255</td>
<td>255</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>255</td>
<td>0</td>
<td>255</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>255</td>
<td>208</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>255</td>
<td>0</td>
<td>255</td>
<td>219</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>255</td>
<td>0</td>
<td>215</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>255</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>207</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>255</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>204</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Actual image value is 210
Case (iv) Pₗ is corrupted and less than 2 uncorrupted pixels in the selected 7 X 7 window

\[
\begin{array}{ccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 255 & 255 & 255 & 0 & 255 \\
0 & 0 & 0 & 255 & 0 & 255 & 0 \\
255 & 0 & 0 & 255 & 0 & 255 & 255 \\
0 & 0 & 255 & 0 & 255 & 0 & 255 \\
0 & 255 & 255 & 0 & 0 & 0 & 0 \\
255 & 255 & 0 & 0 & 255 & 255 & 255 \\
\end{array}
\]

Consider the 3 x 3 window around Pₗ with already processed 4 pixels

\[
\begin{array}{ccc}
104 & 103 & 102 \\
101 & 255 & 0 \\
255 & 0 & 255 \\
\end{array}
\]

Min value : 101 ; Max value : 104
Midpoint value = (101+104)/2 = 102.5
Pₗ replaced by 102.5

\[
\begin{array}{ccc}
104 & 103 & 102 \\
101 & 102.5 & 0 \\
255 & 0 & 255 \\
\end{array}
\]

Actual image value is 103

2.3 SIMULATION RESULTS

The performance of the MAUTMPF is tested with different gray scale images. MF, AMF, PSMF, DBA, MDBA and MDBUTMF algorithms are used for comparison. The MAUTMPF and other filters used for
comparison are implemented in MATLAB 7.8 on a PC equipped with 2.6 GHz CPU and 2 GB RAM memory for the evaluation. The noise density is varied from 10% to 95%. De-noising performances are quantitatively measured by PSNR, MSE, SSIM and IEF defined in Equation (1.2) to Equation (1.5) respectively.

The PSNR and MSE values of the MAUTMPF and previous researches for varying ND from 10% to 90% for lena image are shown in Table 2.1 and Table 2.2 respectively. A graphical representation of SSIM and IEF measures are shown in Figure 2.18 and Figure 2.19 respectively for the same input Lena image.

Table 2.1 Comparison of PSNR values of proposed MAUTMPF and previous approaches for lena image at different NDs

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>PSNR in dB</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ND 10%</td>
<td>ND 20%</td>
<td>ND 30%</td>
<td>ND 40%</td>
<td>ND 50%</td>
<td>ND 60%</td>
<td>ND 70%</td>
<td>ND 80%</td>
</tr>
<tr>
<td>MF</td>
<td>26.34</td>
<td>25.66</td>
<td>21.86</td>
<td>18.21</td>
<td>15.04</td>
<td>11.08</td>
<td>9.93</td>
<td>8.68</td>
</tr>
<tr>
<td>AMF</td>
<td>28.43</td>
<td>27.40</td>
<td>26.11</td>
<td>24.40</td>
<td>23.36</td>
<td>20.60</td>
<td>15.25</td>
<td>10.31</td>
</tr>
<tr>
<td>PSMF</td>
<td>30.22</td>
<td>28.39</td>
<td>25.52</td>
<td>22.49</td>
<td>19.13</td>
<td>12.10</td>
<td>11.84</td>
<td>8.02</td>
</tr>
<tr>
<td>DBA</td>
<td>36.40</td>
<td>32.90</td>
<td>30.15</td>
<td>28.49</td>
<td>26.52</td>
<td>24.41</td>
<td>21.47</td>
<td>20.44</td>
</tr>
<tr>
<td>MDBA</td>
<td>36.94</td>
<td>33.11</td>
<td>30.41</td>
<td>29.43</td>
<td>27.54</td>
<td>25.61</td>
<td>22.47</td>
<td>21.32</td>
</tr>
<tr>
<td>MDBUTMF</td>
<td>37.91</td>
<td>34.78</td>
<td>32.29</td>
<td>30.32</td>
<td>28.18</td>
<td>26.43</td>
<td>24.30</td>
<td>21.70</td>
</tr>
<tr>
<td>MAUTMPF</td>
<td>42.53</td>
<td>39.50</td>
<td>37.99</td>
<td>36.56</td>
<td>35.37</td>
<td>34.13</td>
<td>32.55</td>
<td>30.70</td>
</tr>
</tbody>
</table>

From the PSNR values shown in Table 2.1, it is seen that the MAUTMPF demonstrates PSNR improvement of 16.1 dB, 11.9 dB, 12.5 dB, 7.8 dB, 7.6 dB and 5.7 dB for ND of 30% and 22.6 dB, 17.3 dB, 20.7 dB, 11.1 dB, 10.1 dB and 8.3 dB for 70% ND compared to MF, AMF, PSMF, DBA, MDBA and MDBUTMF algorithms respectively.
It is noticed that from Table 2.2, that MAUTMPF exhibits significant MSE reduction of 6.89, 14.34, 28.05, 46.06, 80.00, 122.81, 205.44, 384.26 and 840.90 for NDs from 10% to 90% respectively when compared to MDBUTMF, which shows better results among MF, AMF, PSMF, DBA and MDBA when tested against lena image. From the results, it is seen that the MAUTMPF demonstrates better PSNR improvement and MSE reduction compared to the existing algorithms irrespective of the level of noise densities.

Table 2.2 Comparison of MSE values of proposed MAUTMPF and previous approaches for lena image at different NDs

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ND 10%</th>
<th>ND 20%</th>
<th>ND 30%</th>
<th>ND 40%</th>
<th>ND 50%</th>
<th>ND 60%</th>
<th>ND 70%</th>
<th>ND 80%</th>
<th>ND 90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>MF</td>
<td>151.00</td>
<td>177.00</td>
<td>424.00</td>
<td>982.00</td>
<td>2037.00</td>
<td>5071.00</td>
<td>6608.00</td>
<td>8812.00</td>
<td>14063.00</td>
</tr>
<tr>
<td>AMF</td>
<td>93.00</td>
<td>118.00</td>
<td>159.00</td>
<td>326.00</td>
<td>500.00</td>
<td>566.00</td>
<td>1941.00</td>
<td>6055.00</td>
<td>10473.00</td>
</tr>
<tr>
<td>PSMF</td>
<td>62.00</td>
<td>94.00</td>
<td>182.00</td>
<td>367.00</td>
<td>794.00</td>
<td>4009.00</td>
<td>4257.00</td>
<td>10258.00</td>
<td>14325.00</td>
</tr>
<tr>
<td>DBA</td>
<td>14.90</td>
<td>33.30</td>
<td>62.80</td>
<td>92.10</td>
<td>144.90</td>
<td>235.50</td>
<td>463.50</td>
<td>587.60</td>
<td>1140.50</td>
</tr>
<tr>
<td>MDBA</td>
<td>13.15</td>
<td>31.77</td>
<td>59.17</td>
<td>74.14</td>
<td>114.57</td>
<td>178.68</td>
<td>368.20</td>
<td>479.82</td>
<td>979.67</td>
</tr>
<tr>
<td>MDBUTMF</td>
<td>10.52</td>
<td>21.63</td>
<td>38.38</td>
<td>60.41</td>
<td>98.87</td>
<td>147.94</td>
<td>241.59</td>
<td>439.62</td>
<td>939.90</td>
</tr>
<tr>
<td>MAUTMPF</td>
<td>3.63</td>
<td>7.29</td>
<td>10.33</td>
<td>14.35</td>
<td>18.87</td>
<td>25.13</td>
<td>36.15</td>
<td>55.36</td>
<td>99.00</td>
</tr>
</tbody>
</table>

The SSIM and IEF values of lena image processed by MAUTMPF and other existing algorithms are graphically represented in Figure 2.2 and Figure 2.3 respectively. From Figure 2.2 and Figure 2.3 it is seen that proposed MAUTMPF algorithm outperforms the existing algorithms in terms of SSIM and IEF especially at very high NDs. The comparative results of SSIM for lena image shown in Figure 2.2 reveals that the SSIM values of existing algorithms starts to degrade drastically when ND exceeds 50%, whereas MAUTMPF performs very well even at 90% ND.
Figure 2.2 Comparison of SSIM values of output lena image processed by MAUTMPF and other existing algorithms for various NDs

Figure 2.3 Comparison of IEF values of output lena image processed by MAUTMPF and existing algorithms for various NDs

Several standard images such as lena, pepper, baboon, boat and bridge are used to evaluate the qualitative performance of MAUTMPF algorithm against the nature of input image. The output of MAUTMPF algorithm for different test
images viz., lena, pepper, baboon, boat and bridge for NDs varying from 10% to 95% are shown in Figure 2.4 to Figure 2.8 respectively.

![Figure 2.4: Output lena images processed by MAUTMPF algorithm](image-url)
Figure 2.5 Output pepper images processed by MAUTMPF algorithm
Figure 2.6 Output baboon images processed by MAUTMPF algorithm
Figure 2.7 Output boat images processed by MAUTMPF algorithm
Figure 2.8 Output bridge images processed by MAUTMPF algorithm
The subjective visualization results shown in Figure 2.4 to Figure 2.8 reveals that the proposed MAUTMPF filter can remove salt and pepper noise from the corrupted images effectively while preserving edges and image details very well at various NDs irrespective of the nature of the input image. Also note that the filter performs ex-ordinarily well even for very high noise densities up to 95%. In addition, the PSNR values of proposed MAUTMPF and algorithms used for comparison for two different noise densities viz., 30% and 70% with Lena, Pepper, Baboon, Boat and Bridge input images are shown in Table 2.3. From Table 2.3, it is seen that the MAUTMPF algorithm outperforms all other previous algorithms in terms of PSNR and MSE irrespective of the nature of image both at low and high NDs.

Table 2.3 Comparison of PSNR values of MAUTMPF and existing algorithms for lena, pepper, baboon, boat and bridge images at NDs 30% and 70%

<table>
<thead>
<tr>
<th>ND</th>
<th>Image</th>
<th>MF</th>
<th>AMF</th>
<th>PSMF</th>
<th>DBA</th>
<th>MDBUTMF</th>
<th>Proposed MAUTMPF</th>
</tr>
</thead>
<tbody>
<tr>
<td>30%</td>
<td>Lena</td>
<td>21.86</td>
<td>26.11</td>
<td>25.52</td>
<td>30.15</td>
<td>30.41</td>
<td>32.29</td>
</tr>
<tr>
<td></td>
<td>Pepper</td>
<td>19.26</td>
<td>31.82</td>
<td>31.25</td>
<td>32.86</td>
<td>31.25</td>
<td>31.59</td>
</tr>
<tr>
<td></td>
<td>Boat</td>
<td>17.57</td>
<td>28.05</td>
<td>29.64</td>
<td>27.84</td>
<td>28.99</td>
<td>30.05</td>
</tr>
<tr>
<td></td>
<td>Bridge</td>
<td>20.46</td>
<td>27.52</td>
<td>29.06</td>
<td>31.28</td>
<td>31.31</td>
<td>32.33</td>
</tr>
<tr>
<td>70%</td>
<td>Lena</td>
<td>9.93</td>
<td>15.25</td>
<td>11.84</td>
<td>21.47</td>
<td>22.47</td>
<td>24.30</td>
</tr>
<tr>
<td></td>
<td>Pepper</td>
<td>9.52</td>
<td>18.72</td>
<td>14.56</td>
<td>17.02</td>
<td>21.87</td>
<td>25.24</td>
</tr>
<tr>
<td></td>
<td>Baboon</td>
<td>12.8</td>
<td>15.24</td>
<td>15.23</td>
<td>16.56</td>
<td>18.88</td>
<td>21.37</td>
</tr>
<tr>
<td></td>
<td>Boat</td>
<td>12.8</td>
<td>15.24</td>
<td>15.23</td>
<td>16.56</td>
<td>18.88</td>
<td>21.37</td>
</tr>
</tbody>
</table>

The qualitative performance of the MAUTMPF algorithm is evaluated by observing the visual quality of the images processed by MAUTMPF and algorithms used for comparison for different test images viz., lena, pepper, baboon, boat and bridge images for two different noise densities (low – 30% and high – 70%) and are shown in Figure 2.9 to Figure 2.18.
Figure 2.9 Output lena images processed by MAUTMPF and existing algorithms for 30% noise density
Figure 2.10 Output lena images processed by MAUTMPF and existing algorithms for 70% noise density
Figure 2.11 Output pepper images processed by MAUTMPF and existing algorithms for 30% noise density
Figure 2.12 Output pepper images processed by MAUTMPF and existing algorithms for 70% noise density
Figure 2.13 Output baboon images processed by MAUTMPF and existing algorithms for 30% noise density
Figure 2.14 Output baboon images processed by MAUTMPF and existing algorithms for 70% noise density.
Figure 2.15 Output boat images processed by MAUTMPF and existing algorithms for 30% noise density
Figure 2.16 Output boat images processed by MAUTMPF and existing algorithms for 70% noise density
Figure 2.17 Output bridge images processed by MAUTMPF and existing algorithms for 30% noise density
Figure 2.18 Output bridge images processed by MAUTMPF and existing algorithms for 70% noise density
Note that, from the output images shown in Figure 2.9 to Figure 2.18, the visual quality of the restored images using MAUTMPF is better compared to the output images processed by all other algorithms used for comparison. Also it is observed that the edges and fine details of the output image processed by MAUTMPF are well preserved when compared to the output images processed by other existing algorithms.

The visual quality of the all the images processed by MF, AMF and PSMF for NDs above 60% shows poor quality, however the performance is good for low NDs. The output images processed by DBA, MDBA and MDBUTMF demonstrates better visual quality, however they fail to suppress the impulse noise at higher NDs above 70%.

The subjective quality of the output of MAUTMPF and other existing filters for lena image corrupted by 90% SPN is shown in Figure 2.19. It is observed that the MAUTMPF exhibits better visual quality even at very high ND and MAUTMPF shows better performance in preserving edges and fine details of the image.

The clarity of the output images is validated by observing a zoomed portion of Lena image processed by MAUTMPF and other existing algorithms for 60% ND, which is shown in Figure 2.20. It is seen from Figure 2.20 the output image processed by MAUTMPF have better visual quality compared to the output images processed by other existing algorithms.
Figure 2.19 Output lena images processed by MAUTMPF and existing algorithms for 90% noise density
Further to measure the effectiveness of the proposed MAUTMPF algorithm both at low and high NDs, a graphical representation of SSIM values of the output lena, pepper, baboon, boat and bridge images processed by MAUTMPF and other existing algorithms for 30% and 70% NDs are shown in Figure 2.21 and Figure 2.22 respectively and IEF values for 30% and 70% NDs are shown in Figure 2.23 and Figure 2.24 respectively.
Figure 2.21 Comparison of SSIM values of MAUTMPF and existing algorithms for lena, pepper, baboon, boat and bridge images at 30 % ND

It is seen from Figure 2.21 and Figure 2.22, the SSIM values of MAUTMPF is similar when compared to the DBA, MDBA and MDBUTMF approaches at 30% ND and significantly better with SSIM value being 38.4%, 35.7%, 15.2%, 24.7% and 23.1% higher for lena, pepper, baboon, boat and bridge images respectively, when compared to the best of the designs used for comparison at 70% ND.

It is noticed from Figure 2.23 and Figure 2.24, that the IEF values of the proposed MAUTMPF is improved by 19.2%, 7.6%, 9.7%, 16.6% and 67.2% at 30% ND and 67.1%, 69.5%, 29.7%, 58.2% and 79.5% at 70% ND for lena, pepper, baboon, boat and bridge images respectively, when compared to the best of the designs used for comparison.
Figure 2.22 Comparison of SSIM values of MAUTMPF and existing algorithms for lena, pepper, baboon, boat and bridge images at 70 % ND

Figure 2.23 Comparison of IEF values of MAUTMPF and existing algorithms for lena, pepper, baboon, boat and bridge images at 30 % ND
Figure 2.24 Comparison of IEF values of MAUTMPF and existing algorithms for lena, pepper, baboon, boat and bridge images at 70 % ND

The qualitative performance of MAUTMPF for lena color images is presented in Figure 2.25 for 70% noise density. From the visual observation, it is clear that MAUTMPF outperforms all other existing algorithms.
2.4 CONCLUSION

In this chapter an efficient de-noising algorithm viz., MAUTMPF for digital images corrupted by high impulse noise is proposed. Experimental evaluation of the MAUTMPF algorithm using standard test images viz., lena, pepper, baboon, boat and bridge demonstrates better performance in terms of PSNR, MSE, SSIM and IEF values compared to previous algorithms.
In addition, the performance test of MAUTMPF algorithm against varying noise densities also revealed superior performance against prior algorithms both at high and low noise densities irrespective of the nature of input image.