Block Matching and 3D Filtering for Image Denoising

Abhinit Bajaj¹, Sukhjit Singh²
¹M.Tech (Student), ECE Section, GTBKIET Chhapianwali, Malout, INDIA
²Assistant Professor, ECE Section, GTBKIET Chhapianwali, Malout, INDIA
Punjab Technical University, Jalandher, INDIA

ABSTRACT
We present a novel approach to still image denoising based on effective filtering in 3D transform domain by combining sliding-window transform processing with block-matching. We process blocks within the image in a sliding manner and utilize the block-matching concept by searching for blocks which are similar to the currently processed one. The matched blocks are stacked together to form a 3D array and due to the similarity between them, the data in the array exhibit high level of correlation. We exploit this correlation by applying a 3D decorrelating unitary transform and effectively attenuate the noise by shrinkage of the transform coefficients. The subsequent inverse 3D transform yields estimates of all matched blocks. After repeating this procedure for all image blocks in sliding manner, the final estimate is computed as weighed average of all overlapping block estimates.

A fast and efficient algorithm implementing the proposed approach is developed. The experimental results show that the proposed method delivers state-of-art denoising performance, both in terms of objective criteria and visual quality.

Keywords: Image Denoising, Block-matching, 3D transforms.

I. INTRODUCTION

In these methods we undertake the block-matching concept for a single noisy image; as we process image blocks in a sliding manner, we search for blocks that exhibit similarity to the currently-processed one. The matched blocks are stacked together to form a 3D array. In this manner, we induce high correlation along the dimension of the array in which the blocks are stacked. We exploit this correlation by applying a 3D decorrelating unitary transform which produces a sparse representation of the true signal in 3D transform domain. Efficient noise attenuation is done by applying a shrinkage operator (e.g. hard thresholding or Wiener filtering) on the transform coefficients. This results in improved denoising performance and effective detail preservation in the local estimates of the matched blocks, which are reconstructed by an inverse 3D transform of the filtered coefficients. After processing all blocks, the final estimate is the weighted average of all overlapping local block-estimates. Because of overcompleteness which is due to the overlap, we avoid blocking artifacts and further improve the estimation ability.

II. DENOISING BY SHRINKAGE IN 3D TRANSFORM DOMAIN WITH BLOCK-MATCHING

Let us introduce the observation model and notation used throughout the chapter. We consider noisy observations \( z : X \rightarrow R \) of the form
\[
z(x) = y(x) + \eta(x), \quad x \in X\]
where \( x \in X \) is a 2D spatial coordinate that belongs to the image domain \( X \), \( y \) is the true image, and \( \eta(x) \) is white Gaussian noise of zero mean and variance \( \sigma^2 \). By \( Z_x \) we denote a block of fixed size \( N \times N \) extracted from \( z \), which has \( z(x) \) as its upper-left element; alternatively, we say that \( Z_x \) is located at \( x \). With \( \hat{y} \) we designate the final estimate of the true image.

A- Local Estimates

We successively process all overlapping blocks of fixed size in a sliding manner, where "process" stands for the consecutive application of block-matching and denoising in local 3D transform domain. For the subsections to follow, we fix the currently processed block as \( Z_{xR} \), where \( x_R \in X \), and denote it as "reference block".

B- Block-matching

Block-matching is employed to find blocks that exhibit high correlation to \( Z_{xR} \). Because its accuracy is significantly impaired by the presence of noise, we utilize a block-similarity measure which performs a coarse initial denoising in local 2D transform domain. Hence, we define
a block-distance measure (inversely proportional to similarity) as
\[
d(Z_{x_1}, Z_{x_2}) = N_1^{-1} \| T_{2D}(Z_{x_1}), \lambda_\text{thr,2D} \|_2 \left( \sqrt{2\log(N_1^2)} \right)
\]
where \( x_1, x_2 \in X \), \( T_{2D} \) is a 2D linear unitary transform operator (e.g. DFT), \( \gamma \) is a hard-threshold operator, \( \lambda_\text{thr,2D} \) is fixed threshold parameter, and \( \| \|_2 \) denotes the \( L^2 \)-norm. Naturally, \( \gamma \) is defined as
\[
\gamma(\lambda, \lambda_\text{thr}) = \begin{cases} 
\lambda, & \text{if } |\lambda| > \lambda_\text{thr} \\
0, & \text{otherwise}
\end{cases}
\]
The result of the block-matching is a set \( S_{xR} \subseteq X \) of the coordinates of the blocks that are similar to \( Z_{xR} \) according to our \( d \)-distance (1); thus, \( S_{xR} \) is defined as
\[
S_{xR} = \{ x \in X \mid d(Z_{xR}, Z_x) < \tau_{\text{match}} \}, \quad (2)
\]
where \( \tau_{\text{match}} \) is the maximum \( d \)-distance for which two blocks are considered similar. The matching procedure in presence of noise is demonstrated on Figure 1, where we show a few reference blocks and the ones matched as similar to them.

\[\text{Figure 1} \quad \text{Fragments of Lena, House, Boats and Barbara corrupted by AWGN of } \sigma = 15. \text{ For each fragment block matching is illustrated by showing a reference block marked shown with red color and a few of its matched ones.}\]

\[\text{C- Denoising in 3D transform domain}\]

We stack the matched noisy blocks \( Z_{xSxR} \) (ordering them by increasing \( d \)-distance to \( Z_{xR} \)) to form a 3D array of size \( N_1 \times N_1 \times |S_{xR}| \), which is denoted by \( Z_{SxR} \). We apply a unitary 3D transform \( T_{3D} \) on \( Z_{SxR} \) in order to attain sparse representation of the true signal. The noise is attenuated by hard-thresholding the transform coefficients. Subsequently, the inverse transform operator \( T_{3D}^{-1} \) yields a 3D array of reconstructed estimates
\[
\hat{Y}_{SxR} = T_{3D}^{-1}\left( T_{3D}(Z_{SxR}), \lambda_\text{thr,3D} \|_2 \left( \sqrt{2\log(N_1^2)} \right) \right) \quad (3)
\]
where \( \lambda_\text{thr,3D} \) is a fixed threshold parameter. The array \( \hat{Y}_{SxR} \) comprises of \( |S_{xR}| \) stacked local block estimates \( \hat{Y}_{xR} \) of the true image blocks located at \( x \in S_{xR} \). We define a weight for these local estimates as
\[
w_{xR} = \begin{cases} 
1, & \text{if } N_{\text{har}} \geq 1 \\
1, & \text{otherwise}
\end{cases} \quad (4)
\]
where \( N_{\text{har}} \) is the number of non-zero transform coefficients after hard-thresholding.

\[\text{D- Estimate Aggregation}\]

After processing all reference blocks, we have a set of local block estimates \( \hat{Y}_{xR} \) (and their corresponding weights \( w_{xR} \)), which constitute an overcomplete representation of the estimated image due to the overlap between the blocks. The final estimate \( \hat{y} \) is computed as a weighted average of all local ones as given by
\[
\hat{y}(x) = \frac{\sum_{xR \in X} \sum_{xSxR} w_{xR}\hat{Y}_{xR}(x)}{\sum_{xR \in X} \sum_{xSxR} w_{xR}} \quad (5)
\]
where \( X_{sm} \) is the characteristic function of the square support of a block located at \( x_m \).

\[\text{III. WIENER FILTER EXTENSION}\]

Provided that an estimate of the true image is available (e.g. it can be obtained from the method given in the previous section), we can construct an empirical Wiener filter as a natural extension of the above thresholding technique. Because it follows the same approach, we only give the few fundamental modifications that are required for its development and thus omitting repetition of the concept. Let us denote the initial image estimate by \( e : X \rightarrow R \). In accordance with our established notation, \( E_x \) designates a square block of fixed size \( N1 \times N1 \), extracted from \( e \) and located at \( x \in X \).

\[\text{A- Modification to Block-Matching}\]

In order to improve the accuracy of block-matching, it is performed within the initial estimate \( e \) rather than the noisy image. Accordingly, we replace the thresholding based \( d \)-distance measure from (1) with the normalized \( L^2 \)-norm of the difference of two blocks with subtracted means. Hence, \( S_{xR} \) becomes
\[
S_{xR} = \{ x \in X \mid \| (E_{xR} - \bar{E}_{xR}) - (E_x - \bar{E}_x) \|_2 < \tau_{\text{match}} \} \quad (6)
\]
where \( \bar{E}_{xR} \) and \( \bar{E}_x \) are the mean values of the blocks \( E_{xR} \) and \( E_x \), respectively. The mean subtraction allows for improved matching of blocks with similar structures but different mean values.
**B- Modification to Denoising in 3D Transform Domain**

The linear Wiener filter replaces the nonlinear hard-thresholding operator. The attenuating coefficients for the Wiener filter are computed in 3D transform domain as

$$W_{SR} = \frac{|T_{3D}(E_{SR})|^2}{|T_{3D}(E_{SR})|^2 + \sigma^2},$$

where $E_{SR}$ is a 3D array built by stacking the matched blocks. We filter the 3D array of noisy observations $Z_{SR}$ in $T_{3D}$-transform domain by an elementwise multiplication with $W_{SR}$. The subsequent inverse transform gives

$$\hat{Y}_{SR} = T_{3D}^{-1}(W_{SR}T_{3D}(Z_{SR})), \quad (7)$$

where $\hat{Y}_{SR}$ comprises of stacked local block estimates of the true image blocks located at the matched locations. As in (4), the weight assigned to the estimates is defined as

$$W_{xR} = \left(\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sum_{t=1}^{N_3} |W_{SR}(i,j,t)|^2\right)^{-1}, \quad (8)$$

**IV. ALGORITHM**

We present an algorithm which employs the hard-thresholding approach to deliver an initial estimate for the Wiener filtering part that produces the final estimate. A straightforward implementation of this general approach is computationally demanding. Thus, in order to realize a practical and efficient algorithm, we impose constraints and exploit certain expedients.

It is often assumed that neighboring pixels in small blocks extracted from natural images exhibit high correlation; thus, such blocks can be sparsely represented by well-established decorrelating transforms, such as the DCT, the DFT, wavelets, etc. From computational efficiency point of view, however, very important characteristics are the separability and the availability of fast algorithms. Hence, the most natural choice for $T_{2D}$ and $T_{3D}$ is a fast separable transform which allows for sparse representation of the true-image signal in each dimension of the input array.

**Efficient Image Denoising Algorithm with Block-Matching and 3D Filtering**

Let us introduce constraints for the complexity of the algorithm. First, we fix the maximum number of matched blocks by setting an integer $N_2$ to be the upper bound for the cardinality of the sets $S_{xR}$. Second, we do block-matching within a local neighbourhood of fixed size $N_3 \times N_3$ centered about each reference block, instead of doing it in the whole image. Finally, we use $N_{step}$ as a step by which we slide to every next reference block. Accordingly, we introduce $X_R \subseteq X$ as the set of the reference blocks’ coordinates, where $|X_R| \approx \frac{|X|}{N_{step}}$.

In order to reduce the impact of artifacts on the borders of blocks (border effects), we use a Kaiser window $W_{win2D}$ (with a single parameter $\beta$) as part of the weights of the local estimates. These artifacts are inherent of many transforms (e.g. DFT) in presence of sharp intensity differences across the borders of a block.

Let the input noisy image be of size $M \times N$, thus $|X| = MN$. We use two buffers of the same size, $ebuff$ for estimates and $wbuff$ for weights to represent the summations in the numerator and denominator, respectively, in (5.5). For simplicity, we extend our notation so that $e_{buffer}(x)$ denotes a single pixel at coordinate $x$ and $ebuff_x$ designates a block located at $x$ in $ebuff$ (the same notation is to be used for $wbuff$). A flowchart of the hard-thresholding part of the algorithm is given in Figure 2 (but we do not give such for the Wiener filtering part since it requires only the few changes.). Following are the steps of the image denoising algorithm with block-matching and 3D filtering.

(i). **Initialization.** Initialize $e_{buffer}(x) = 0$ and $wbuff(x) = 0$, for all $x \in X$.

(ii). **Local hard-thresholding estimates.** For each $x_R \in X_R$, do the following sub-steps.
(a) Block-matching. Compute $S_{SR}$ as given in Equation (2) but restrict the search to a local neighbourhood of fixed size $N_S \times N_S$ centred about $x_R$. If $|S_{SR}| > N_S^2$, then let only the coordinates of the $N_S^2$ blocks with smallest $d$-distance to $Z_{SR}$ remain in $S_{SR}$ and exclude the others.

(b) Denoising by hard-thresholding in local 3D transform domain. Compute the local estimate blocks $\hat{Y}_{xSR}$ and their corresponding weight $w_{xR}$ as given in (3) and (5.4), respectively.

(c) Aggregation. Scale each reconstructed local block estimate $\hat{Y}_{xSR}$, by a block of weights $W(x_R) = w_{xR}$ $W_{win2D}$ and accumulate to the estimate buffer: $e_{buff} = e_{buff} + W(x_R) \hat{Y}_{xSR}$, for all $x \in S_{SR}$. Accordingly, the weight block is accumulated to the same locations as the estimates but in the weights buffer: $w_{buff} = w_{buff} + W(x_R)$, for all $x \in S_{SR}$.

(iii). Intermediate estimate. Produce the intermediate estimate $e(x) = \frac{e_{buff}(x)}{w_{buff}(x)}$ for all $x \in X$, which is to be used as initial estimate for the Wiener counterpart.

(iv). Local Wiener filtering estimates. Use $e$ as initial estimate. The buffers are re-initialized: $e_{buff}(x) = 0$ and $w_{buff}(x) = 0$, for all $x \in X$. For each $x_R \in X_R$, do the following sub-steps.

(a) Block-matching. Compute $S_{SR}$ as given in (6) but restrict the search to a local neighbourhood of fixed size $N_S \times N_S$ centred about $x_R$. If $|S_{SR}| > N_S^2$, then let only the coordinates of the $N_S^2$ blocks with smallest $d$-distance to $E_{SR}$ remain in $S_{SR}$ and exclude the others.

(b) Denoising by Wiener filtering in local 3D transform domain. The local block estimates $\hat{Y}_{xSR}$ and their weight $w_{xR}$ are computed as given in (7) and (8), respectively.

(c) Aggregation. It is identical to step (ii)c.

(v). Final estimate. The final estimate is given by $\hat{y}(x) = \frac{e_{buff}(x)}{w_{buff}(x)}$, for all $x \in X$.

V. RESULTS AND DISCUSSION

We present experiments conducted with the algorithm introduced in Section 4, where the transforms $T_{2D}$ and $T_{3D}$ are the 2D DFT and the 3D DFT, respectively. All results are produced with the same fixed parameters—but different for the hard-thresholding and Wiener filtering parts. We consider two sample images Lena (512x512 pixels) and Cameraman (256x256 pixels) as shown below for proposed method $SADCT[]$ and $K$-$SVD[]$. We summarize the results of the proposed technique in terms of output peak signal-to-noise ratio (PSNR) in decibels (dB), in Table 1 and Table 2 for Lena and Cameraman image respectively. The diagrams below also shows their respective graphs and reconstructed images. This is used to compare the relative filtering performance of various methods. The PSNR between the filtered output image $K(i,j)$ and the original image $I(i,j)$ of dimension $M \times N$ pixels is defined as:

$$\text{PSNR} = 10 \cdot \log_{10} \left( \frac{\text{MAX}_I^2}{\text{MSE}} \right) = 20 \cdot \log_{10} \left( \frac{\text{MAX}_I}{\sqrt{\text{MSE}}} \right)$$

Here, MAX is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255.

MSE (mean square error) is defined as:

$$\text{MSE} = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

<table>
<thead>
<tr>
<th>METHODS</th>
<th>BM3D-DFT</th>
<th>SA-DCT</th>
<th>K-SVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>STANDARD DEVIATION OF GAUSSIAN NOISE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>43.52</td>
<td>43.33</td>
<td>42.46</td>
</tr>
<tr>
<td>5</td>
<td>38.63</td>
<td>38.54</td>
<td>38.52</td>
</tr>
<tr>
<td>10</td>
<td>35.82</td>
<td>35.58</td>
<td>35.47</td>
</tr>
<tr>
<td>15</td>
<td>34.21</td>
<td>33.86</td>
<td>33.6</td>
</tr>
<tr>
<td>20</td>
<td>33.03</td>
<td>32.62</td>
<td>32.28</td>
</tr>
<tr>
<td>25</td>
<td>32.06</td>
<td>31.66</td>
<td>31.33</td>
</tr>
<tr>
<td>40</td>
<td>29.41</td>
<td>29.17</td>
<td>28.7</td>
</tr>
<tr>
<td>50</td>
<td>28.68</td>
<td>28.59</td>
<td>27.69</td>
</tr>
<tr>
<td>75</td>
<td>27.01</td>
<td>26.75</td>
<td>25.8</td>
</tr>
<tr>
<td>100</td>
<td>26.04</td>
<td>25.49</td>
<td>24.46</td>
</tr>
</tbody>
</table>
Table 1: PSNR Values for denoised Lena image versus standard deviation of AWGN of noisy image for BM3DFT, SADCT and KSVD.

<table>
<thead>
<tr>
<th>STANDARD DEVIATION OF GAUSSIAN NOISE</th>
<th>BM3D-DFT</th>
<th>SAD-DCT</th>
<th>K-SVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>43.74</td>
<td>43.54</td>
<td>43.14</td>
</tr>
<tr>
<td>5</td>
<td>37.99</td>
<td>37.15</td>
<td>37.01</td>
</tr>
<tr>
<td>10</td>
<td>33.76</td>
<td>33.98</td>
<td>32.08</td>
</tr>
<tr>
<td>15</td>
<td>31.5</td>
<td>31.7</td>
<td>31.52</td>
</tr>
<tr>
<td>20</td>
<td>30.34</td>
<td>30.18</td>
<td>30.02</td>
</tr>
<tr>
<td>25</td>
<td>29.08</td>
<td>29.11</td>
<td>28.79</td>
</tr>
<tr>
<td>40</td>
<td>26.97</td>
<td>26.9</td>
<td>26.16</td>
</tr>
<tr>
<td>50</td>
<td>25.55</td>
<td>25.38</td>
<td>25.12</td>
</tr>
<tr>
<td>75</td>
<td>24.29</td>
<td>23.87</td>
<td>23.62</td>
</tr>
<tr>
<td>100</td>
<td>22.73</td>
<td>22.49</td>
<td>22.32</td>
</tr>
</tbody>
</table>

Graph 1: PSNR Values for denoised image versus standard deviation of AWGN for Lena image for BM3DFT, SADCT and KSVD

Reconstructed Images For Lena

(a) (b) (c) (d)

Figure 4: Lena images (a) noisy image with standard deviation of AWGN 20, denoised images with (b) KSVD (c) SADCT (d) BM3DFT.

Table 2: PSNR Values for denoised Cameraman image versus standard deviation of AWGN of noisy image for BM3DFT, SADCT and KSVD.

PSNR values for Cameraman Image

Graph 2: PSNR Values for denoised image versus standard deviation of AWGN for Cameraman image for BM3DFT, SADCT and KSVD

Reconstructed Images for Cameraman

(a) (b)
We conclude by remarking that the proposed method outperforms—in terms of objective criteria—all techniques known to us. Moreover, our estimates retain good visual quality even for relatively high levels of noise. Our current research extends the presented approach by the adoption of variable-sized blocks and shape-adaptive transforms, thus further improving the adaptivity to the structures of the underlying image. Also, application of the technique to more general restoration problems is being considered.

REFERENCES


