Artifact reduction of compressed images and video combining adaptive fuzzy filtering and directional anisotropic diffusion

Nadernejad, Ehsan; Forchhammer, Søren; Korhonen, Jari

Published in:
2011 3rd European Workshop on Visual Information Processing (EUVIP)

Link to article, DOI:
10.1109/EuVIP.2011.6045551

Publication date:
2011

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
ARTIFACT REDUCTION OF COMPRESSED IMAGES AND VIDEO COMBINING
ADAPTIVE FUZZY FILTERING AND DIRECTIONAL ANISOTROPIC DIFFUSION

Ehsan Nadernejad, Søren Forchhammer, and Jari Korhonen
Dept. of Photonics Engineering, Technical Univ. of Denmark, Kgs. Lyngby, Denmark

ABSTRACT
Fuzzy filtering is one of the recently developed methods for reducing distortion in compressed images and video. In this paper, we combine the powerful anisotropic diffusion equations with fuzzy filtering in order to reduce the impact of artifacts. Based on the directional nature of the blocking and ringing artifacts, we have applied directional anisotropic diffusion. Besides that, the selection of the adaptive threshold parameter for the diffusion coefficient has also improved the performance of the algorithm. Experimental results on JPEG compressed images as well as MJPEG and H.264 compressed videos show improvement in artifact reduction of the proposed algorithm over other directional and spatial fuzzy filters.

Index Terms—Artifact Reduction, Anisotropic Diffusion, Fuzzy filter, H.264.

1. INTRODUCTION
Image and video compression suffers from spatial and temporal distortions. Spatial distortion includes blocking and ringing, and typical temporal distortion types are mosquito and flickering artifacts. Blocking artifacts are caused by separate compression of each block, and it occurs both in horizontal and vertical direction of each frame. Ringing artifacts occur when the high frequency transform coefficients obtained from Discrete Cosine Transform (DCT) or wavelet-based coding are quantized or truncated. This causes ripples or oscillations around sharp edges or contours in the image. It is also known as Gibbs Phenomenon. When ringing artifacts are changing from frame to frame as a video sequence is displayed, mosquito artifacts are created. Flickering artifacts [7] appear due to the inconsistency in quality at the same spatial position in adjacent frames.

Over the years, many algorithms have been proposed to reduce the spatial and temporal artifacts. For de-blocking, linear low-pass filters in [1,2] and Projection Onto Convex Sets “POCS” [3] have been used, but these methods cause blurriness and have a high computational complexity. To remove ringing artifacts, linear or non-linear isotropic filters can be applied to the regions near to edges [4, 5]. For combating flickering artifacts, most of the current methods focus on reducing the flickering in intra-frame coding [7,8].

In [7], the quantization error is considered when obtaining the optimal intra prediction mode and to help reducing the flickering. Also in [8], flickering is included in the cost function, when the optimal prediction mode and block-size are chosen. In [9,10], spatiotemporal fuzzy filters are used to remove different artifacts.

In image processing and computer vision, anisotropic diffusion, also called Perona-Malik diffusion, is a technique aiming at reducing the noise without removing essential parts of the image content, such as edges, lines and other details that are important for the interpretation of the image [11,16].

Fuzzy filters are improved median filters or a rank condition rank selection filters [14], where the binary spatial-rank relation is replaced by a real valued relation. This permits the filter to adapt to the spread of the signal by averaging the flat areas, while the isolated pixels in the edge areas remain.

In this paper, we propose a new algorithm using anisotropic diffusion processing and spatial fuzzy filtering [9,10] to reduce coding artifacts in compressed images and video. We process all the vertical and horizontal artifacts using one-dimensional (1D) anisotropic diffusion after applying the fuzzy filter. To avoid the blurring effect, anisotropic diffusion is performed with a small number of iterations.

The rest of the paper is organized as follows: Section 2 summarizes the anisotropic diffusion. Section 3 provides a description of fuzzy filtering. Adaptive fuzzy filtering and anisotropic diffusion for reducing artifacts in compressed video sequences are combined in Section 4. Section 5 shows the simulation results and compares the proposed algorithm with known methods. Finally, the concluding remarks are given in Section 6.

2. ANISOTROPIC DIFFUSION
Diffusion is a fundamental physical process. The isotropic diffusion process can be modeled as a Gaussian smoothing process, where its variance increases continuously. For the anisotropic diffusion, the smoothing process may be performed differently in each direction. Let \( I(x,y,t) \) represent an image at the coordinates \((x, y)\) at time \( t \) of the diffusion, when the diffusion flux is defined as:

\[
J(x,y,t) = -D(x,y,t) \nabla I(x,y,t)
\]
\[ \varphi = -C \nabla I \]  

(1)

where \( C \) is the diffusion coefficient. With the matter continuity equation we get:

\[ \frac{\partial I}{\partial t} = -\nabla \cdot \varphi \]  

(2)

By combining (1) and (2), the diffusion equation is obtained:

\[ \frac{\partial I}{\partial t} = \nabla \cdot (C \nabla I) \]  

(3)

where “\( \cdot \)” represents the inner product of two vectors. When \( C \) is a constant parameter, the diffusion process is isotropic. When \( C \) is a function of the directional parameters, the diffusion process becomes anisotropic. To solve the partial differential equation (3), the image \( I_0 \) is used as the initial condition and the Neumann boundary condition is applied to the image borders:

\[ I(x,y,t)_{|t=0} = I_0, \quad \partial_n I = 0 \]  

(4)

The discrete form of the diffusion equation is given by:

\[ \frac{\partial I(x,y,t)}{\partial t} = \nabla \cdot (C(x,y,t)\nabla I(x,y,t)) \]  

(5)

where \( C \) is the diffusion coefficient and \( I \) is the original image in each time scale \( t \).

Perona-Malik (PM) [11] suggested the two well-known diffusion coefficients (6) and (7):

\[ C_i(s) = \frac{1}{1 + \left(\frac{s}{k}\right)^2} \quad (6) \]

\[ C_i(s) = \exp\left[-\left(\frac{s}{k}\right)^2\right] \quad (7) \]

where \( s = |\nabla I| \). By using diffusion equations, the diffusion process will be prominent when the magnitude of the local gradient is low, and will be resisted when the magnitude of the local gradient is high.

In both (6) and (7), parameter \( k \) is a factor controlling the diffusion process. Both equations provide perceptually similar results, but (6) emphasizes noise removal, while (7) emphasizes high contrast preservation.

In some cases, the PM method can misinterpret noise as edges and create false edges by enhancing the noise. Therefore, Catte et al. [12] changed \( s = |\nabla I| \) in the PM diffusion function to:

\[ s = |\nabla G_\sigma \ast I| \]  

(8)

Here, \( G_\sigma \) is a Gaussian smoothing kernel and “\( \ast \)” is the convolution operator. In this approach, \( |\nabla G_\sigma \ast I| \) is used to estimate the local gradient instead of the more noise-sensitive \( |\nabla I| \).

A new method is introduced to find parameter \( k \) in diffusion coefficient (6,7). The idea is based on calculating the gradient of the image in different directions. The experiments on different images show that this parameter depends on the changes of the image gradient in different directions,

\[ k = \alpha \cdot \nu \]  

(9)

where, \( \alpha \) is a constant and \( \nu \) is the variance of the image gradient in different directions. By choosing \( \alpha = 2 \cdot 10^{-6} \) the optimal results were obtained. The parameter was calculated based on experiments on 100 standard images from [17].

3. FUZZY FILTER

Fuzzy filters improve the median filters or rank condition rank selection filters [10, 13] by replacing the binary spatial-rank relation by a real-valued relation. In [14], fuzzy filter is defined by generalizing the binary spatial-rank relation. Assuming that filter \( h \) is applied to a set \( \omega = \{1,2,\ldots,M\} \) of neighboring samples \( f[i+j,f] \) around the input \( f[i,j] \), we can formulate the output:

\[ g[i,j] = \frac{\sum_{|i+j|=0} h(f[i+i',j+j'], f[i,j]) \times f[i+i',j+j']} {\sum_{|i+j|=0} h(f[i+i',j+j'], f[i,j])} \]  

(10)

And its unbiased form via normalization:

\[ g[i,j] = \frac{\sum_{|i+j|=0} h(f[i+i',j+j'], f[i,j]) f[i+i',j+j']} {\sum_{|i+j|=0} h(f[i+i',j+j'], f[i,j])} \]  

(11)

where \( h(f[i+i',j+j]', f[i,j]) \) represents the input of the filter \( f[i+i',j+j]' \) to the output.

Due to the input independence of the filter coefficients, a low-pass filter designed to perform effectively in the flat areas may introduce blurring artifacts in the detailed areas. However, it is desirable to preserve the details, while removing the artifacts. This can be achieved by imposing constraints, such as if \( f[i+i',j+j]' \) is far from \( f[i,j] \), its contribution to the output is small. In this case, the filter coefficients \( h[i,j] \) must follow the constraints:

\[ \lim_{|i+j|=0} \frac{h(f[i+i',j+j'], f[i,j])}{f[i,j]} = \frac{1}{1} \]  

(13)

\[ \lim_{|i+j|=0} \frac{h(f[i+i',j+j'], f[i,j])}{f[i,j]} = \frac{0}{1} \]  

(14)

\[ h(f[i+i',j+j'], f[i,j]) \geq h(f[i+i',j+j'], f[i,j]) \]  

(15)

The function \( h(f[i+i',j+j'], f[i,j]) \) is called a membership function, and there are many functions that fulfill these requirements. A Gaussian membership function is given in:

\[ h(f[i+i',j+j'], f[i,j]) = \exp\left(\frac{(f[i+i',j+j'] - f[i,j])^2}{2\sigma^2}\right) \]  

(16)

where \( \sigma \) represents the spread parameter of the input and controls the strength of the fuzzy filter. The input \( x[i,j] \) contributes the output always more that the other samples:

\[ h(f[i,j], f[i,j]) \geq h(f[i+i',j+j'], f[i,j]) \forall k \]  

(17)

For the same \([f[i+i',j+j'], f[i,j]]\), the higher the value of \( \sigma \), the higher the contribution of \([f[i+i',j+j'], f[i,j]]\) to the output. This implies that \( f[i,j] \) will converge more towards \( f[i+i',j+j]' \). Smaller values of \( \sigma \) will keep the signal \( f[i,j] \) more isolated from its neighboring samples. The spread parameter should be adaptive to different areas with different activity levels, such as smooth or detailed textures. The conventional fuzzy filter uses fixed spread parameters for every surrounding sample, ignoring their relative positions. In image and video compression, distortions such as blocking, ringing or flickering artifacts are directional, and, thus, the direction
between \(f[i,j]\) and its surrounding samples \(f[i+1,j+1]\) should be taken into consideration. This can be achieved by an adaptive spread parameter:

\[
\sigma(f[i+i',j+j'], f[i,j]) = K[i,i',j,j'] \times \sigma(t[i,j])
\]  

where \(\sigma_t\) is a position-dependent amplitude function of the spread parameter, and \(K\) is the scaling function controlled by the direction of \([i+i', j+j']\) to \([i,j]\). In this paper, we use cosine-based general form for the spread parameter (19):

\[
\sigma(\theta) = \sigma_t(\alpha + \beta \cos^2(\theta)),
\]

where \(\sigma_t\) is a constant, \(\theta\) is the direction between the pixel positions \([i,j]\) and \([i+i', j+j']\), \(\alpha\) and \(\beta\) are positive scaling factors controlling the maximum and minimum strength of the filter. The extensions of the membership function \(\sigma\) in (18) are discussed in [9,10] for compressed images and compressed video sequences. More information about fuzzy filters is available in [5,13,14].

4. PROPOSED METHOD FOR REMOVING ARTIFACTS

In the proposed method, adaptive 1-D fuzzy filtering [10] is first applied to the pixels suffering from blocking artifacts, and then directional anisotropic diffusion is used to increase the quality. Finally, adaptive 2-D fuzzy filtering [9] is applied to the pixels with ringing artifacts (see Fig. 1).

![Flowchart of the proposed method](image)

Fig. 1 Flowchart of the proposed method.

In the first phase, vertical artifact detection is performed along each vertical boundary of an 8x8 block. For this purpose, the difference between each pair of boundary pixels is computed (see Fig. 2), and if \(\text{MAX}(L1,L2,L3,L4) < G0\), or \(\text{MAX}(R1,R2,R3,R4) < G0\), the current row is marked as a boundary gap and is filtered by a 1D fuzzy filter. Similar analysis is performed also horizontally.

![Detect the vertical boundary gap in a row across the block vertical boundary](image)

Fig. 2 Detect the vertical boundary gap in a row across the block vertical boundary.

At the block classification step, 2D fuzzy filter is adapted to the variation of the pixels in each block. This is based on standard deviation (STD) of blocks (See Fig. 3).

\[
\text{STD}(I(t,m,n)) = \sqrt{\frac{1}{5} \sum_{t=1}^{5} \sum_{m=1}^{5} I(t,k,l) - \frac{1}{9} \sum_{p=1}^{9} \sum_{q=1}^{9} I(t,p,q)}
\]

The classification is based on the value of the maximum \(\text{STD}(\text{MaxSTD})\) in an 8x8 block, as shown in (21).

\[
\begin{align*}
(45,<> &\Rightarrow \text{StrongEdge} \\
(25,25) &\Rightarrow \text{WeakEdge} \\
(5,15) &\Rightarrow \text{WeakTexture} \\
(0,5) &\Rightarrow \text{Smooth}
\end{align*}
\]

where \(\text{MaxSTD}\) is the maximum \(\text{STD}\) of the pixels in each block.

![Calculating Standard Deviation](image)

Fig. 3 Calculating Standard Deviation

To avoid smoothing and increase the quality of the images, 1D anisotropic diffusion is performed instead of two-dimensional diffusion. Also, as mentioned in Section 2, the proposed anisotropic diffusion is an adaptive algorithm.

The numerical solution of the proposed algorithm (Adaptive Anisotropic Diffusion) is described below.

1. Find the value of \(\delta\) parameter based on (9).
2. Let the time step be \(\Delta t\) and the spatial step be \(h\) in \(x, y\) directions. Then, the time and space coordinates can be presented in discrete form as:

\[
i = n\Delta, n = 0,1,2,\ldots; x = ih, y = jh,
\]

\[
i = 0,1,2,3,\ldots M - 1, j = 0,1,2,\ldots N - 1
\]

where \(Mh \times Nh\) is the size of the image. Let \(I^0_{i,j} = I(ih, jh, n\Delta t)\) then the final image can be obtained using the four-stage approach described below:

1. Stage I: The horizontal derivative approximations and the horizontal Laplacian approximations are computed for the frame:

\[
\nabla^2 I^0_{i,j}(\text{hor}) = (I^0_{i+1,j} + I^0_{i-1,j} - 2 \times I^0_{i,j})
\]

The symmetric boundary conditions are used:

\[
I^0_{0,j} = I^0_{M,j}, I^0_{M,j} = I^0_{0,j}, j = 0,1,2,\ldots ,N - 1
\]

2. Stage II: Computing the directional diffusion coefficient \(c(x,y;\tau)\), as for example:

\[
c^x_{i,j}(\tau) = \exp\left(\frac{-\nabla^2 I^0_{i,j}(\text{hor})}{2\tau^2}\right)
\]

The other diffusion coefficients can be obtained in a similar fashion.

3. Stage III: Computing the divergence of \(c(\cdot)\nabla \vec V\):

\[
d^+_{i,j} = \frac{1}{h^2} [c^+_{i+1,j}(I^0_{i+1,j} - I^0_{i,j}) + c^+_{i-1,j}(I^0_{i-1,j} - I^0_{i,j})]
\]

With the symmetric boundary conditions:

\[
d^+_{0,j} = d^+_{M,j}, d^+_{M,j} = d^+_{0,j}, j = 0,1,2,\ldots ,N - 1
\]

4. Stage IV: The numerical approximation to the differential equation is given by:

\[
I^{n+1}_{i,j}(\text{hor}) = I^0_{i,j}(\text{hor}) + \Delta t 16 d^+_{i,j}(\text{hor})
\]

This equation is equivalent to (5) (in this work: \(\Delta t/16 = 0.125\)). The same algorithm is used in the vertical direction.
At the following step, 2D fuzzy filter is applied to each block based on MaxSTD. Figure 4 displays the flowchart of the criteria for selection of fuzzy parameters.

![Flowchart](image)

Fig. 4 Selection of adaptive spread parameter \( \sigma \).

In this algorithm (Fig. 1), the fuzzy filter is used to reduce blocking and ringing artifacts. The fuzzy filter helps reducing the artifacts while retaining the sharpness of the edges. The main drawback of this fuzzy filter is that it is isotropic for multi-dimensional signals while artifacts are directional. To avoid blurring, directional adaptive anisotropic diffusion is applied using few iterations.

To remove artifacts in video coded sequences H.264, a new algorithm is proposed (see Fig. 5). In this algorithm, first, the chroma components are up-sampled to the same size of the luma components. To obtain higher quality, each Y frame is enhanced by a directional spatial fuzzy filter and each U and V frame is de-noised in the same way as the algorithm in Fig.1.

![Diagram](image)

Fig. 5. Proposed method for video sequence

### 5. SIMULATION RESULTS

We tested the processing of MJPEG and H.264 decoded images of test video sequences. We have used two well known metrics to evaluate the algorithm. Peak Signal to Noise Ratio (PSNR) and Mean Structural Similarity (MSSIM) index are used for images, and Weighted-PSNR and Weighted-MSSIM [15] are used for the video coded sequences (i.e. H.264).

#### 5.1 Enhancement of Compressed Images

Simulations have been performed to demonstrate performance of the proposed method. Different approaches are compared in terms of visual quality: PSNR and MSSIM. For the comparison, the methods proposed in [9,10,13] have been implemented (2D fuzzy (spatial), directional fuzzy filter and isotropic fuzzy filter). Only the non-edge pixels (with \( G > 180 \) in [9]) are filtered to avoid destroying the real edges of the image. All the parameters in Section 5 are chosen experimentally using a range of sequences in order to achieve the best visual quality.

For the directional fuzzy filter, parameter \( \sigma \) in (19) is chosen to effectively remove the overall artifacts [9]. Parameter \( \gamma \) controls the balance between removing the artifacts in the flat regions and keeping the details in the high activity regions. Parameters \( \alpha \) and \( \beta \) are used to adjust the relative filtering strength between the gradient and tangent directions of the edges. These parameters are defined in [9], and we have determined their values experimentally, resulting in \( \sigma_i = 15 \), \( \alpha = 0.5 \), and \( \beta = 3.5 \) in (19) and \( \gamma = 0.5 \) as defined in [9]. In the proposed method, parameter \( \sigma \) is chosen based on Fig. 4. The set \( \omega \) of neighboring pixels and the spatial window \( W \) size are set to 5x5. Several CIF resolution video sequences have been compressed using motion JPEG with scaling factor 4 for the quantization step. The test set used includes different types of images from the following video sequences: Silent, Foreman, Mobile, Paris, News, and Mother (50 frames used from each sequence).

To demonstrate the visual quality, results obtained with different artifact removal techniques on a compressed and zoomed frame in Mobile are shown in Fig. 6. For this simulation, the spread parameter has been calculated using the method in Fig. 4. Compared to the compressed full frame in Fig. 6b (PSNR=21.253db, MSSIM=0.482), the enhanced image using 2D fuzzy filter in Fig. 6c (PSNR=21.775, MSSIM=0.482), the enhanced image using the proposed method in Fig. 6d (PSNR=21.961, MSSIM=0.491) achieved a clear improvement in terms of PSNR and MSSIM. Comparing to the compressed image in Fig 6b, the 2D fuzzy method can remove most of the blocking and ringing artifacts. However, it introduces other artifacts, such as blurring at the dominant edges and loss of details. The frame enhanced with the proposed algorithm achieves better quality, compared to the other fuzzy methods. It efficiently removes the blocking and ringing artifacts, but still keeps the details and the sharpness of the edges.

The average PSNR and MSSIM values for different sequences are listed in Tables 1 and 2, respectively. The average improvement in PSNR and MSSIM when using the proposed method, Isotropic [13], Directional [10], and 2D Fuzzy [9] are (0.821db, 0.076), (0.301db, 0.044), (0.608db, 0.056) and (0.633db, 0.062).

<table>
<thead>
<tr>
<th>Sequence</th>
<th>PSNR JPEG</th>
<th>2D Fuzzy</th>
<th>Directional</th>
<th>Isotropic</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>News</td>
<td>27.671</td>
<td>28.091</td>
<td>27.855</td>
<td>27.961</td>
<td>28.265</td>
</tr>
<tr>
<td>Silent</td>
<td>27.321</td>
<td>27.812</td>
<td>27.985</td>
<td>27.536</td>
<td>28.164</td>
</tr>
<tr>
<td>Mother</td>
<td>30.811</td>
<td>31.824</td>
<td>31.652</td>
<td>31.512</td>
<td>31.871</td>
</tr>
</tbody>
</table>

Table 1: Comparison of PSNR in dB for Different methods
Ave Gain 0.633 0.608 0.301 0.821

Table 2: Comparison of MSSIM for different methods

<table>
<thead>
<tr>
<th>Method</th>
<th>MSSIM JPEG</th>
<th>2 D Fuzzy</th>
<th>Directiona l</th>
<th>Isotropic</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>News</td>
<td>0.532</td>
<td>0.554</td>
<td>0.546</td>
<td>0.542</td>
<td>0.560</td>
</tr>
<tr>
<td>Silent</td>
<td>0.512</td>
<td>0.556</td>
<td>0.543</td>
<td>0.521</td>
<td>0.578</td>
</tr>
<tr>
<td>Foreman</td>
<td>0.460</td>
<td>0.482</td>
<td>0.490</td>
<td>0.475</td>
<td>0.491</td>
</tr>
<tr>
<td>Mobile</td>
<td>0.534</td>
<td>0.541</td>
<td>0.538</td>
<td>0.529</td>
<td>0.551</td>
</tr>
<tr>
<td>Paris</td>
<td>0.484</td>
<td>0.518</td>
<td>0.506</td>
<td>0.495</td>
<td>0.528</td>
</tr>
<tr>
<td>Ave Gain</td>
<td><strong>0.662</strong></td>
<td><strong>0.656</strong></td>
<td><strong>0.644</strong></td>
<td><strong>0.676</strong></td>
<td><strong>0.660</strong></td>
</tr>
</tbody>
</table>

Fig. 6. JPEG zoomed images for comparison: (a) Original Frame (b) Compressed; (c) 2D Fuzzy (d) Proposed Method

5.2 Enhancement of Compressed Video Sequences

To evaluate the proposed method for video sequences (Fig.5), different methods have been applied to decoded MJPEG and H.264 sequences. In the MJPEG codec, each frame is compressed separately using the JPEG standard compression. For practical reasons, the scaling function in (18) is set as a constant, $K[i+i',j+j'] = 1$, and $\sigma = \sigma_1$ is given by Fig. 4. The sizes of $\omega$ and the spatial window are 5x5 pixels.

Figures 7 and 8 compare the PSNR and MSSIM values of the methods tested for 50 frames of the Mobile sequence. The plots clearly demonstrate that the proposed Fuzzy-PDE filter achieves consistent PSNR gain of about 0.53dB on average, relative to the compressed frames, and about 0.12dB compared to the frames enhanced with the conventional fuzzy spatial filter method [10]. The respective MSSIM gains are about 0.02 and 0.05.

The visual improvement obtained with the proposed scheme is much more noticeable when the processed frames are played in a sequence, as the proposed method produces a video of smoother quality with significantly reduced artifacts.

In order to demonstrate that the proposed method is beneficial also for more efficient video compression methods, further experiments have been performed using H.264 video compression. The Foreman sequence was compressed with the prediction structure of IPPP at a bitrate of 132 Kbps. Two different types of experiments have been performed on the H.264 video sequences. In the first experiment, the in-loop de-blocking filter was disabled. In the second experiment, the algorithm was applied with in-loop filtering enabled. The adaptive spread parameter $\sigma$ is applied to each block and the offset $\gamma$ [9] was set to 0.5. These parameters have been chosen experimentally to get the best visual quality for a wide range of sequences.

In the first experiment, the proposed algorithm improves PSNR and MSSIM (averages are 34.912 db, 0.9834), compared to PSNR and MSSIM with fuzzy method (averages are 34.623 db, 0.9812) and disable in-loop filtering (averages are 34.248 db, 0.979).

In the second experiment, the proposed method improves PSNR and MSSIM (averages are 34.881 db, 0.9864), compared to PSNR and MSSIM with the fuzzy method (averages are 34.766 db, 0.9814) and in-loop filtering (averages are 34.38 db, 0.979). This improvement is consistent for the Foreman sequence, which is verified by the PSNR and MSSIM curves shown in Figs. 9 and 10. Visual analysis shows that compared to in-loop filtering, the fuzzy method and the proposed method both significantly reduce the ringing artifacts close to edges. However, the proposed method maintains the sharpness of edges better than the fuzzy method. This improvement indicates a more pleasant visual appearance.

Fig. 7. Comparison on PSNR of simulated methods.

Fig. 8. Comparison on MSSIM of simulated methods.

The enhanced video sequence using the proposed method has fewer artifacts than the compressed sequence enhanced with the fuzzy method and the in-loop de-blocking filter. The PSNR improvement with the proposed algorithm applied to the Foreman sequence compressed with different bitrates is shown in Fig. 11. The proposed algorithm yields more than 0.33 dB improvement, compared to the fuzzy
method giving 0.21 dB improvement for bitrates from 70 Kbps to 170 Kbps. Also, visual analysis indicates higher quality with the proposed method.

Another benefit of our method is that it does not require a motion compensation stage and spatiotemporal filtering. This is why its computational complexity is reasonably low.

6. CONCLUSIONS

In this paper, we have proposed an effective algorithm for image and video artifact removal using an adaptive fuzzy filter and directional anisotropic diffusion. This novel method overcomes the limitations of the conventional nonlinear filters by taking pixel’s activity and the direction between pixels both into account. It has been shown that the proposed algorithm improves the visual quality of compressed images and videos in terms of PSNR and MSSIM, compared to existing approaches. The proposed adaptive scheme can be applied to different image and video compression standards, such as JPEG, MJPEG and H.264.

8. REFERENCES