TECHNICAL TRANSACTIONS | CZASOPISMO TECHNICZNE
AUTOMATIC CONTROL | AUTOMATYKA
2-AC/2013

VOLODYMYR FEDAK, ADRIAN NAKONECHNY*

SPATIO-TEMPORAL ALGORITHM FOR CODING ARTIFACTS REDUCTION IN HIGHLY COMPRESSED VIDEO

PRZESTRZENNO-CZASOWY ALGORYTM REDUKCJI ARTEFAKTÓW W WYSOCE SKOMPRESOWANYM FILMIE

A b s t r a c t

Images and video are often coded using block-based discrete cosine transform (DCT) or discrete wavelet transform (DWT) which cause a great deal of visual distortions. Restoration of image sequences can obtain better results compared to restoring each image individually, provided that the temporal redundancy is adequately used. In this article, efficient approach for artifacts reduction has been presented. In order to enhance the overall video quality, the proposed approach uses image sequence redundancy. Spatial and temporal information is used for the video de-noising process.

Keywords: *artifacts, spatio-temporal postprocessing, wavelet transformation*

S t r e s z c z e n i e

Zdjęcia i filmy są często kodowane za pomocą blokowej dyskretnej transformaty kosinusowej (DCT) lub dyskretnej transformaty falkowej (DWT), które powodują duże zakłócenia wizualne. Przywrócenie sekwencji obrazów pozwala uzyskać lepsze wyniki niż przywracanie każdego obrazu osobno, pod warunkiem odpowiedniego użycia redundancji czasowej. W niniejszym artykule zaprezentowano efektywne podejście do redukcji artefaktów. W celu zwiększenia ogólnej jakości obrazu omawiane podejście wykorzystuje redundancję sekwencji obrazu. Do procesu odszumiania filmu wykorzystano informacje czasowe i przestrzenne.

Słowa kluczowe: *artefakty, obróbka przestrzenno-czasowa, transformacja falkowa*

* M.Sc. Volodymyr Fedak, e-mail: vfedak@softserveinc.com, Prof. D.Sc. Ph.D. Andrian Nakonechnyy, Department of Computerized Automatic Systems, Lviv Polytechnic National University.

1. Introduction

Every time we need to obtain, process, and deliver information. This information is not limited to text files or sample messages, nevertheless various visual pieces of information could be transmitted including image and video files. However, transmission channels have limited bandwidth and storage devices have a limited capacity. Digital video is broadcast and stored in an encoded form, so it requires less information (bits) than the original. At low bit-rates, the coarse quantization exploited during compression results in visually annoying coding artifacts [1].

Compression artifact is a particular class of data errors that are usually the consequence of quantization in lossy data compression. These distortions can be classified into the following types:

Blocking artifacts. Such types of image/video distortions are the most visible degradation of all artifacts. This effect is caused by all block-based coding techniques. It is a well-known fact that all compression techniques divide the image into small blocks and then compress them separately. Due to the coarse quantization, the correlation among blocks is lost, and horizontal and vertical borders appear.

Ringing artifacts. The ringing effect is caused by the quantization or truncation of the high frequency coefficients and can also come from improper image restoration operations. Ringing artifacts are visible for all compression techniques especially when image or video is transformed into frequency domain. Moreover, it appears as distortion along sharp edges in the video sequence. This artifact occurs very often when the DWT encoder is used. Furthermore, it may be observed after the image or video has been de-coded using a frequency coder.

Blur effect. Blurring is another artifact resulting from the absence of high frequencies in low bit rate video. It appears around the sharp edges and all image details become blurred. This effect is very similar to the ringing artifact, and sometimes it is hard to distinguish between them.

Flickering is one of the most annoying temporal artifacts that appears in video. As it is widely known, modern algorithms encode video as a sequence of images. The first frame from this sequence is a key frame (I), others are additional (previous [P] and subsequent [B]) frames. All sequences are encoded by motion-compensated algorithms. When an observer watches the de-coded video, the flickering effect is noticeable due to the difference between key frames (I) and other frames (P, B).

Different techniques could be used to reduce most annoying artifacts and all of these techniques could be divided by filtering domain (spatial, frequency, temporal). Different authors provide versatile methods of image/video quality improvements and sometimes the most challenging task is to chose the necessary technique. However, the most promising results are shown by patch-based methods that use image/video self-similarity for the artifacts reduction task.

In general, all post-processing methods (that use an image/video redundancy) could be divided into two types: those that use temporal information; those that only use spatial information. Having several images of the same scene can be greatly beneficial to the restoration results. The first step in exploiting temporal redundancy is inferring the connection between the images. This connection is what sets apart treating an image sequence from treating a random set of images. The connection is usually inferred by estimating the motion between the frames in the sequence, i.e., detecting the location of each pixel in each image in every other image [3]. In some cases, when the motion is of global nature (e.g., an affine transform), it is relatively simple to accurately estimate the motion trajectories. However, most sequences contain very complex motion patterns of non-rigid shapes and with many occlusions. In such cases, motion estimation is a severely under-determined problem, and is very prone to errors and inaccuracies. Several recent algorithms developed for the denoising of the image sequences show that example-based methods that are able to bypass the classic explicit motion estimation need [4–6]. Spatial filtering is commonly used for noise and artifacts reduction. However, some artifacts, such as temporal flickering or severe blockiness, cannot be removed efficiently using only spatial techniques. In order to remove highly resistant artifacts, information from adjacent frames should be used. Thus, our algorithm relying on the motion estimation attempt to detect areas where the motion estimation is reliable, and turn to spatio-temporal image sequences processing mechanisms for those areas.

In this article, an efficient algorithm based on spatio-temporal filtering for artifacts reduction has been presented. The proposed algorithm can reduce the most annoying effects such as: ‘blockiness’, ‘flickering’ and ‘ringing’. In order to diminish artifacts in video sequences, our approach tries to take advantage of the redundancy and self-similarity of the image sequences. A true motion-estimation algorithm is required to effectively use temporal information. Therefore, one existing motion-estimation algorithm is used and functionality is added to determine the quality of each motion vector.

2. Existing approaches to artifacts reduction

In modern digital systems and video broadcast chains, video compression is applied to reduce bandwidth or storage size. Post-processing of the decoded image sequence is an acceptable technique to achieve a better perceived picture quality [10]. Furthermore, modern consumer vision products like televisions and PCs use image enhancement and restoration techniques to improve the objective and subjective picture quality. All postprocessing algorithms and methods can be divided into the following types [1]:

– Spatial filtering;
– Filtering in the frequency/wavelet domain;
– Temporal filtering;
– Hybrid algorithms (mainly combines spatial and frequency filtering).

Many approaches have been proposed in the literature aimed at the alleviation of the blocking artifacts in the images and video. Spatial algorithms modify image pixel values. These approaches are usually used together with the edge detection algorithms to prevent the blurring effect. As nowadays a great number of algorithms have been developed, it would be rational to overview these approaches due to which completely versatile solutions can be reached.

With the purpose of improving image and video quality authors in [12] proposed the algorithm that uses local statistics of transform coefficients. The authors investigated that pixel brightness diversity among blocks is greater than within one block, and border pixels are filtered by the spatial algorithm. This approach reduces the blocking effect from the image and simultaneously introduces the additional blur to the image’s edges.

In [10], authors used local statistics as a means of differentiation between monotone and edge blocks and introduced a generic filter for the removal of blocking artifacts and the staircase effect. Monotone blocks contain less spatial details than the edge blocks. They propose to use two-dimensional filtering that is applied for monotone blocks and one-dimensional directional filtering for the edge blocks.

A new pixel classification-based approach for the block artifacts reduction has been proposed in [13]. Instead of classifying each block of fixed size to smooth region or edge region, they distinguish each pixel using the binary edge map from the edge detection process. They reduce grid noise in the smooth region using an adaptive filter.

Most encouraging results could be received using the NLM approach [7]. The efficiency of this algorithm is proven in many different areas and this algorithm tries to take advantage of the redundancy and self similarity of the image. This approach will be discussed in the next sections of this article.

Frequency algorithms transform image or video (sequence of images) to frequency domain and modify DCT (Discrete Cosine Transform) or DWT (Discrete Wavelet Transform) coefficients. These approaches are very efficient but of high complexity because image and video signals have to be transformed from spatial to frequency domain and vice versa. Authors in [10] proposed a adaptive algorithm of blocking artifacts reduction in DCT domain. This algorithm performs filtering using following steps:

– The image is divided into edge and monotone areas. A sobel edge detector [14] is used for this purpose;
– Reduce blocking artifacts in non-edge areas. Horizontal and vertical smoothing filters in the spatial domain is used;
– Apply Filter Tao [15] in the edge areas;
– Transform image to the original format. Quantization Constraints.

The effect of averaging the spatially closest pixels can also be achieved in the Fourier domain. The average of the spatially closest pixels is then equivalent to the cancellation of the high frequencies. As the analogous spatial filter, this cancellation leads to the blurring of the image and a Gibbs effect. The optimal filter in the Fourier domain is the Wiener filter which does not cancel the high frequencies but attenuates them all.

In the wavelet domain, the noise is uniformly spread throughout the coefficients, while most of the image information is concentrated in the few largest ones (sparsity of the wavelet representation) [19–20]. The most straight-forward way of distinguishing information from noise in the wavelet domain consists of thresholding the wavelet coefficients. The soft-thresholding filter is the most popular strategy and has been theoretically justified in [21]. They proposes a three steps denoising algorithm:

– The computation of the forward WT;
– The filtering of the wavelet coefficients;
– The computation of the IWT of the result obtained.

Consequently, regarding the three steps denoising algorithm, there are two tools to be chosen: the WT (Wavelet Transform) and the filter. In [22] the UDWT (Undecimated Discrete Wavelet Transform) was used, in [23] the DTCWT (Dual Tree Complex Wavelet Transforms), and in [24] the DWT.

From the first category can be mentioned the hard-thresholding filter that minimizes the Min-Max estimation error and the Efficient SURE-Based Inter-scales Point-wise Thresholding Filter [24], which minimizes the Mean Square Error (MSE). To the second category belong filters obtained by minimizing a Bayesian risk under a cost function, typically a delta cost function (MAP estimation [25]) or the minimum mean squared error [22]. The denoising algorithms proposed in [24] exploit the inter-scale dependence of wavelet coefficients. The method proposed in [22] takes into account the intra-scale dependence of wavelet coefficients as well. The statistical distribution of the wavelet coefficients changes from scale to scale. The coefficients of the WT have a heavy tailed distribution.

In [11], the authors introduced the wavelet-based de-blocking and de-ringing the algorithm for artifacts suppression. Based on a theoretical analysis of the blocking artifacts, the proposed algorithm is able to take into account the statistical characteristic of block discontinuities, as well as the behaviour of wavelet coefficients across scales for different image features to suppress both the blocking and ringing artifacts.

Temporal filtering is used to diminish different types of artifacts based on temporal information. Furthermore, these techniques are very often used with spatial and frequency algorithms (hybrid algorithms).

The provided review of different approaches demostrates the level of variance for different postprocessing algorithms and methods that were proposed by the last decade. And the main task is to chose a right filering approach that provides the most promising results. Non-Local means filtering has proven efficiency and provides the most promisng results [26, 27], that's why it's used in this research. Hovewer it is worth conducting additional research to compare this approach with other most promising wavelet based algorithms.

2.1. Image filtering using Non-Local Means

All image and movie filters which are intended to reduce noise by averaging similar pixels are considered to be neighbourhood filters. Noise reduction can thus be achieved by averaging the pixels which have received the same original grey level value. The NLM algorithm removes the noise while keeping all this meaningful image information. For this purpose, the NLM algorithm tries to take advantage of the redundancy and self-similarity of the image. Most image details occur several times; each small window in a natural image has many similar windows in the same image.

As example see Fig. 1 from [7].

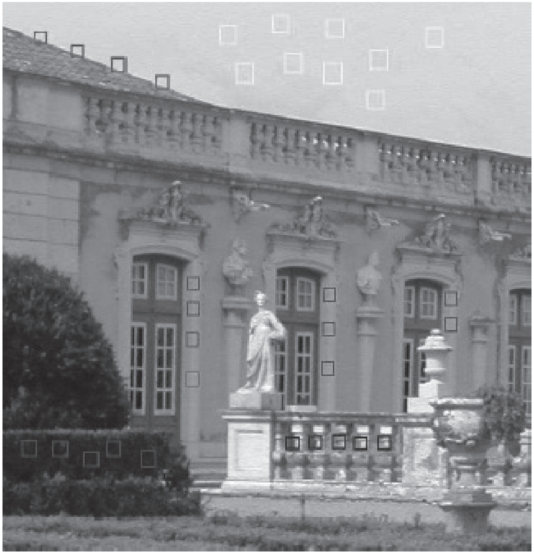

Fig. 1. The similar image patches within the same image. Most image elements appear repeatedly. Each different rectangle indicates a squares in the image which are almost indistinguishable from the set of rectangles with the same color

The NLM algorithm is an improvement for bilateral filtering. The bilateral and the NLM filters are two very successful image de-noising filters. Both the bilateral and the NLM filters are based on the assumption that the image contents are likely to repeat themselves within some neighbourhood. Therefore, the de-noising of each pixel is done by averaging all pixels in its neighbourhood.

The NLM algorithm estimates the value of x as an average of the values of all the pixels. The probability that y is similar to x is determined by looking at the difference in the luminance value and the difference in position between x and y in the neighbourhood filters.

Given a discrete noisy image $v = \{v(i) | i \in I\}$, the estimated value NL(v)(i) is computed as a weighted average of all the pixels in the image:

$$\mathrm{NL}(v)(i) = \sum_{j \in I} w(i, j)\, v(j) \tag{1}$$

The neighbourhood of a pixel x is defined as the set of pixels in a sequence in which each pixel has a surrounding window similar to the window around x. All pixels in this neighbourhood can be used for predicting x. The NLM filter is defined as:

$$\mathrm{NL}_h\{\underline{x}\} = \frac{1}{C(\underline{x})} \cdot \sum_{\underline{y} \in Q(\underline{x})} z(\underline{x}) \cdot e^{-\frac{\left\| \underline{N}(\underline{x}) - \underline{N}(\underline{y}) \right\|_2^2}{h^2}} \tag{2}$$

where:

$C(\underline{x}) = \sum_{\underline{y} \in Q(\underline{x})} e^{-\frac{\left\| \underline{N}(\underline{x}) - \underline{N}(\underline{y}) \right\|_2^2}{h^2}}$	– a normalizing constant,
$N(x)$	– a vector which contains the pixels in the window surrounding pixel x,
$Q(x)$	– a search window around x, in which the neighbourhood of x is searched,
The window $N(x)$	– contains $Sx \cdot Sy$ pixels,
Search window $Q(x)$	– contains $Ax \cdot Ay$ pixels.

Considering the previous research discussed above (all pros. and cons.), our algorithm will need to meet the following requirements:

– As the most encouraged technique for image/video filtering is the algorithms that use image/video redundancy to restore its content, our algorithms should also use redundancy spatial and temporal to restore the image sequence (video) content;
– As algorithm uses information from the temporal domain, the technique must model the motion compensation;
– Information from the temporal domain shouldn't add additional noise;
– To avoid the modification of existing decoders, the post-processor shouldn't use coding parameters.

Based on the specified assumptions and tasks, a new algorithm for artifacts reduction has been developed and presented in this article.

3. Spatio-temporal algorithm for coding artifacts reduction

The NLM filter is based on the assumption that image content is likely to repeat itself within some neighborhood [7]. All pixels that have similar surrounding windows can be used for predicting the luminance of the original scene. Originally, the NLM was designed as the spatial filter. In this way the NLM takes advantage of redundancy that is presented in the spatial domain.

Extension of the NLM to the temporal domain gives more information for the NLM to retrieve the original frame. This algorithm will take advantage over both temporal and spatial domains.

On the one hand, providing more information gives a grater possibility of the NLM retrieving the original frame with higher quality but, on the other hand, it can cause some other undesirable effects. However, this temporal information should be carefully checked before the filtering process. The main goal is not to provide flawed information from the temporal domain, but only to provide useful data. This step can guarantee that no additional noise has been added to the processed frame.

In order to guarantee that no additional noise is added, a true motion estimation algorithm is used for searching motion vectors [2]. This algorithm uses a custom model to verify the quality of each motion vector.

In [18] we presented an original idea/approach of Spatio-Temporal filtering with the motion vectors quality determination. This algorithm was evolved and some part of the initial approach was simplified due to performance reasons. It was determined that for the majority of video signals, 3DRS Motion estimation is good enough and itial step with initial motion vectors finding (based on Gabor vawelets) very rarely has influence to overal filtering.

The general flow chart of the proposed Spatio-temporal algorithm for artifacts reduction is depicted on Fig. 2. This algorithm can be divided into the following steps (additional information about these steps is presented in the next sub-chapters of this article):

– Motion estimation 3DRS. True motion estimation algorithm is used for searching motion vectors [14];
– Determine type of filtering. If motion vector is consistent (error value less than some Threshold), additional temporal information will be used due to it having at an advantage over spatial information, otherwise only temporal information would be used in the filtering process;
– Filtering process. NLM is used as a core algorithm for filtering.

In case motion vector is consistent, additional temporal information will be used which will have advantage over spatial information. If motion vector quality is turned up within the specific fixed range (it is not a final true motion vector but can be used as a temporal candidate), this area will be filtered in the same way as spatial candidates, otherwise (motion vector is not true) only information from the spatial domain will be used [2]. In this implementation, the previous and the next frames are used.

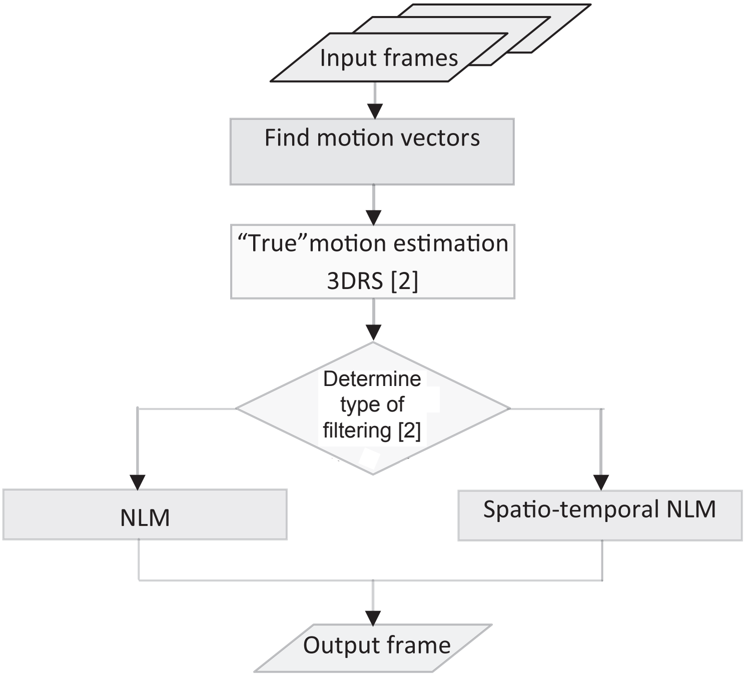

Fig. 2. Flow chart of the spatio-temporal Non-Local Means algorithm

$$\mathrm{NL}_h\{\underline{x}\} = \frac{1}{C(\underline{x})} \cdot \sum_{\underline{y} \in Q(\underline{x})} z(\underline{x}) \cdot e^{-\frac{\|\underline{N}(\underline{x}) - \underline{N}(\underline{y})\|_2^2}{h^2}} \cdot z(\underline{x}) \cdot e^{-\frac{\|\underline{N}(\underline{x}) - \underline{N}_{\pm 1}(\underline{x})\|_2^2}{10 \cdot h^2}} \tag{3}$$

where:

$\underline{N}_{\pm 1}(\underline{x})$ –is the corresponding area of the next frame (+1) or the previous frame (–1).

$$\mathrm{NL}_h\{\underline{x}\} = \frac{1}{C(\underline{x})} \cdot \sum_{\underline{y} \in Q(\underline{x})} z(\underline{x}) \cdot e^{-\frac{\|\underline{N}(\underline{x}) - \underline{N}(\underline{y})\|_2^2}{h^2}} \cdot z(\underline{x}) \cdot e^{-\frac{\|\underline{N}(\underline{x}) - \underline{N}_{\pm 1}(\underline{x})\|_2^2}{h^2}} \tag{4}$$

In case the current block has appropriate patches from the next and the previous frames, two additional patches will be added to corresponding patches from the search area and filtering across all these patches will be performed.

3.1. Find motion vectors between different frames

Motion estimation algorithms calculate the motion between two input images and produce output a field of motion vectors. Block matching is a popular method for estimating motion vectors from the image sequence. It assumes that the motion is uniform over a block of pixels and that the motion can be modelled as the displacement of these blocks. The maximum possible vertical and horizontal displacement of the block defines the search area, and the best matching block is determined by minimizing the Sum of Absolute Difference (SAD) between the source block and the destination block inside the search area.

Plenty of motion estimation algorithms have been proposed [3], among which the three-dimensional-recursive-search (3DRS) has proved to be efficient in many applications [8]. The 3DRS principle is based on the following assumptions:

1. Objects in the frame are assumed to be larger than blocks (block size that is used in motion estimation);
2. Vectors estimated for neighbouring blocks are good candidates for the current block.

To summarize, the candidate vectors are constructed as follows (5) [17]:

$$c_i = \begin{cases} d(x+\rho_i,\ n); \\ d(x+\rho_i,\ n-1); \\ c_j + u, \quad j \neq i, \quad u \in US \end{cases} \tag{5}$$

The candidate set contains two spatial candidates $d(x + \rho, n)$, one temporal candidate $d(x + \rho_i, n - 1)$ and two update candidates $c_j + u$ (Fig. 3).

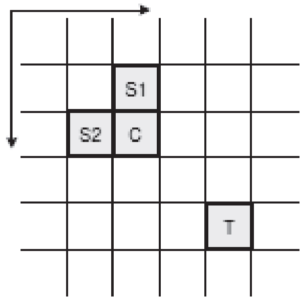

Fig. 3. Candidate set configuration. C is the current block; $S1$, $S2$ and T indicate two spatial and temporal candidates. Update candidates are random candidates generated using $S1$ and $S2$. The arrows indicate the scanning direction

There is one additional requirement which is quite different from other applications like quality of the motions vectors. This parameter is described in more details in the next section.

3.2. Determining motion vectors quality during the process of motion estimation

Motion estimation algorithms calculate the motion between two input images and produce output a field of motion vectors. Block matching is a popular method for estimating motion vectors from image sequences. It assumes that the motion is uniform over a block of pixels and that the motion can be modelled as displacement of these blocks. The maximum possible vertical and horizontal displacement of a block defines the search area, and the best matching block is determined by minimizing the Sum of Absolute Difference (SAD) between the source block and the destination block inside the search area.

There has been lot of motion estimation algorithms developed over the last 2 decades. However it is very difficult to find out the motion vectors quality evaluation for detected motion vectors. The main aim of this section is to highlight the calculating of the motion vectors quality and change the original 3DRS algorithm to be steady for the rapidly changing video.

During the block motion estimation process, each frame is divided into small blocks (in 3DRS implementation, each block has a fixed size of 4 × 4 pixels).

The candidate motion vectors c_i are constructed as follows [17]:

$$c_i = \begin{cases} d(x+\rho_i,\ n) & \text{Spatial candidate} \\ d(x+\rho_i,\ n-1) & \text{Temporary candidate} \\ c_j + u, \quad j \neq i, \quad u \in US & \text{Update motion vector candidate} \end{cases} \tag{6}$$

Sum of Absolute Differences (SAD) with additional penalty (7) is used to determine the best candidate motion vector for the current block [17].

$$p_i = \begin{cases} 1/2 \cdot \beta_0 \cdot \beta_1 & \text{Spatial candidate} \\ \beta_0 \cdot \beta_1 & \text{Temporary candidate} \\ 2 \cdot \beta_0 \cdot \beta_1 & \text{Update motion vector candidate} \end{cases} \tag{7}$$

The candidates set used in this study, contains two spatial candidates, two temporal candidates and four update vectors, and in total, eight candidates per block.

In comparison to the original 3DRS implementation, we added two more update vectors in order to adopt this algorithm for rapidly changing video.

A penalty mechanism ensures preferences for those candidates that have the same displacement. The final motion vector is determined as in the original algorithm 3DRS:

$$d(x,n) = \arg\min(\varepsilon(c_i, x, n) + p_i), \ \ c_i \in \mathrm{CS} \tag{8}$$

The quality of the motion vector can be calculated from the following equation:

$$\mathrm{Quality}(c) = \varepsilon(c_i, x, n) + \alpha \cdot (S)^{1/3} + \beta \cdot \mathrm{dev}(x, y) \in \mathrm{CS} \tag{9}$$

where:

- Quality(c) – is motion vector quality,
- $\varepsilon(c_i, x, n)$ – is a Sum of absolute differences,
- dev(x, y) – deviation of the neighboring blocks,
- α, β – are balancing coefficients.

So, the final NLM filtering type is determined from the next equtation:

$$\mathrm{Quality}(C) = \begin{cases} < \mathrm{Th1} & \text{Motion vector is clear} \\ > \mathrm{Th1}\ \&\&\ < \mathrm{Th2} & \text{Motion vector is within specified range} \\ > \mathrm{Th2} & \text{Motion is inconsistent} \end{cases} \tag{10}$$

where:

Th1, Th2 – Threashold values for quality coefficients of motion vectors.

These values are defines differently based on image/video clour scheme.

4. Results

In this section, an objective analysis of the original NLM algorithm and spatio-temporal algorithm is performed over sequences with several levels of compression (0.5 Mbps, 3 Mbps are used in this research) to evaluate how efficiently the proposed spatio-temporal algorithm reduces the compression artifacts. The proposed spatio-temporal and original NLM algorithms are applied to sequences which are encoded and decoded using the MPEG-2 codec. The blocking and flickering artifacts in these compressed sequences are strongly visible. The MSE and BIM metrics are used for evaluation of the processed sequences. The calculation of MSE and GBIM and the encoding of test sequences are done using the PTS tool.

The Mean Square Error (MSE) is the error metric used to compare image processing (de-noising, compression) quality. The MSE represents the cumulative squared error between the de-noised and the original image. The lower the value of MSE, the higher the quality of a restored signal.

BIM and PSBIM are used to measure the amount of blocking artifacts in the image/video. These metrics show a strong consistency with the human perception of coding impairments and subjective evaluations is the General Block Impairment Metric (GBIM), introduced in [28, 29]. The lower the value of GBIM the lower the quantity of the blocking artifacts. PSBIM is an improved GBIM metric.

Table 1

Objective metrics results for NLM filtering

Sequence	Metric	0.5 Mbps	3 Mbps	Original
Akio	MSE	6.331692	5.646	0
	BIM	1.277975	1.274	7.521864
	PSBIM	0.738614	0.734	1.749776
Bowling	MSE	4.374	4.143	0
	BIM	1.144023	1.1423	8.792035
	PSBIM	0.545938	0.54585	1.465607
Foreman	MSE	39.809	10.215	0
	BIM	1.024102	1.025591	6.129313
	PSBIM	0.758773	0.695938	1.757120
Bus	MSE	264.278	40.115	0
	BIM	1.263775	1.13198	2.051955
	PSBIM	1.124431	1.01635	1.280819
Claire	MSE	3.737	3.167	0
	BIM	1.716750	1.71634	6.548664
	PSBIM	0.999271	0.99699	1.468032

Five different sequences were used in the tests. The sequences were chosen for having varying content and intensity of motion.

Objective metrics for compressed sequences after spatio-temporal NLM filtering are presented in Table 2.

Table 2

Objective metrics results for spatio-temporal NLM filtering

Sequence	Metric	0.5 Mbps	3 Mbps	Original
Akio	MSE BIM PSBIM	4.414 1.261789 0.738614	4.295 1.261 0.734	0 7.521864 1.749776
Bowling	MSE BIM PSBIM	2.814 1.136261 0.530264	2.785 1.13612 0.52991	0 8.792035 1.465607
Foreman	MSE BIM PSBIM	31.295 1.009726 0.746233	10.248 1.02476 0.69826	0 6.129313 1.757120
Bus	MSE BIM PSBIM	234.389 1.229437 1.089562	30.333 1.11699 1.00574	0 2.051955 1.280819
Claire	MSE BIM PSBIM	3.001 1.702442 0.970346	2.835 1.70290 0.96077	0 6.548664 1.468032

For all processed sequences, the MSE values are lower than the MSE of the unprocessed sequences. Sequences processed by spatio-temporal algorithm have slightly less blockiness, meanwhile the BIMs and PSBIMs metrics have slightly better value.

a) b)

Fig. 4. Foreman videosequence: a) processed by NLM, b) processed by proposed spatio-temporal algorithm

Test sequences processed by the spatio-temporal algorithm have lower MSE value, consequently, we received significant improvement and proved that the spatio-temporal algorithm is more effective for highly compressed video. The user can observe essential quality improvements for video processed by means of proposed the spatio-temporal algorithm.

In most cases, proposed the spatio-temporal NLM algorithm can preserve image details in a better way. It is especially visible on highly compressed image/video (Fig. 4a Foreman processed by NLM, 4b by spatio-temporal algorithm).

5. Conclusions

The presented method for artifacts reduction demonstrated that a spatio-temporal approach provides a significant improvement of picture quality at low bitrates compared to spatial filtering only. In case sequences suffering from severe artifacts (e.g. flickering), spatio-temporal filtering proved to be a preferred option. This research also demonstrates benefits that can be achieved by using additional temporal information, especially consistent temporal information (approach for 'consistency' measurement of temporal information also provided in this research).

Temporal filtering is effective mostly for low bitrate videos. High quality sequences simply do not suffer from the severe artifacts propagated to the temporal domain, and therefore do not need much blurring. For those sequences, the most important fact is to differentiate between object details and artifacts, which can be achieved by means of spatial analysis. Therefore, methods based on spatio-temporal analysis and adaptive edge-preserving filtering are the most efficient for high bit-rate videos. Applying spatio-temporal filtering provides better results than original NLM implementation because temporal and spatial information is included in this filtering. Temporal information doesn't propagate additional blurring effects because this information is used only when true motion vector exists. These additional steps to the original NLM algorithm introduce additional complexity, so performance of the proposed algorithm should be enhanced.

True motion estimation is also very important, as it is an initial attempt to introduce some metric for calculating a quality of motion vectors. This technique can be used in other post-processing algorithms or in different directions of image and video processing.

The work described in this article demonstrates advantages of the spatio-temporal filtering approach over the spatial approach and additionally proves the temporal information usage for the coding artifacts reduction in video.

References

[1] Fedak V.I, Nakonechny A.Y., *Artifacts suppression in images and video. Non-Local Means as algorithm for reducing image and video distortions*, Hluboka nad Vltavou, Czech Republic 2009.

[2] Fedak V., Nakonechny A., *Determining motion vectors quality during the process of motion estimation*, 6th International Scientific and Technical Conference 16–19 November 2011, Lviv.

[3] Callico, G.M. et al., *Analysis of fast block matching motion estimation algorithms for video super-resolution systems*, IEEE Transactions on Consumer Electronics, 54(3), 2008, 1430-1438.

[4] Protter M., *Processing Image Sequences Without Motion Estimation*, Technion-Computer Science Department, Ph.D. Thesis.

[5] Boulanger J., Kervrann C., Bouthemy P., *Space-time adaptation for patch based image sequence restoration*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, 2007, 1096-1102.

[6] Dabov K., Foi A., Egiazarian K., *Video denoising by sparse 3D transform-domain collaborative filtering*, Proc. of the European Signal Processing Conference, EUSIPCO, 2007.

[7] Buades A., *Image and film denoising bynon-local means*, Journal of Visual Commincation and Image Representation, Vol. 1, No. 1, Ph. D. Thesis, 24 March 2004, Article No. VC970378.

[8] Ramamurthi B., Gersho A., *Nonlinear space-variant postprocessing of block coded images*, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 34, No. 5, Oct. 1986, 1258,1268.

[9] Truong Quang Vinh, Young-Chul Kim, *Block Artifact Reduction Based on Pixel Classification Using Binary Edge Map*, 2008.

[10] Wang C. et al., *Adaptive Reduction of Blocking Artifacts in DCT Domain for Highly Compressed Images*, 2003.

[11] Alan W.-C. Liew, Hong Yan, IEEE *Blocking Artifacts Suppression in Block-Coded Images Using Overcomplete Wavelet Representation*, IEEE Trans Circuits Syst. Video Technol., Vol. 11, No. 14, No. 4, April 2004.

[12] Choy S.O., Chan Y.-H., Siu W.-C., *Reduction of block-transform image coding artifacts by using local statistics of transform coefficients*, IEEE Signal Processing Letters, Vol. 4, No. 1, Jan. 1997, 5-7.

[13] Truong Quang Vinh, Young-Chul Kim Block., *Artifact Reduction Based on Pixel Classification Using Binary Edge Map* 2008.

[14] Vincent O.R., Folorunso O., A *Descriptive Algorithm for Sobel Image Edge Detection*, Proceedings of Informing Science & IT Education Conference, InSITE, 2009.

[15] Tao C., Wu H.R., Bin Q., *Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts*, IEEE Trans. Circuits Syst. Video Technol., Vol. 11, No. 5, May 2001, 594-602.

[16] Buades A., Coll B., Morel J.-M, *A non-local algorithm for image denoising*, 2006.

[17] De Haan, G. et al., *True-motion estimation with 3-D recursive search block matching*, Circuits and Systems for Video Technology, IEEE Transactions on, 3(5), 1993, 368-379, 388.

[18] Fedak V., Nakonechny A., *Spatio-temporal Non-Local Means Algorithm for Coding Artifacts Reduction*, Lviv Polytechnic National University, 2011.

[19] Zhou Z.-F., Shui P.-L., *Contourlet-based image denoising algorithm using directional windows*, Electronic Letters, 43, No. 2, 2007, 92-93.

[20] Olhede S.C., *Hyperanalytic denoising*, IEEE Transactions on Image Processing, 16 (6), 2007, 1522-1537.

[21] Donoho D.L., Johnstone I.M., *Ideal spatial adaptation by wavelet shrinkage*, Biometrika, Vol. 81, No. 3, 1994, 425-455.

[22] Pizurica A., Philips W., *Estimating the probability of the presence of a signal of interest in multiresolution single and multiband image denoising*, IEEE Transactions on Image Processing, 15, No. 3, 2006, 654-665.

[23] Shui P.-L., *Image Denoising Algorithm via Doubly Local Wiener Filtering With Directional Windows in Wavelet Domain*, IEEE Signal Processing Letters, 12, No. 6, 2005, 681-684.

[24] Luisier F., Blu T., Unser M., *A New SURE Approach to Image Denoising: Inter Scale Orthonormal Wavelet Thresholding*, IEEE Transactions on Image Processing, 16, No. 3, 2007, 593-606.

[25] Achim A., Kuruoglu E.E., *Image Denoising Using Bivariate-Stable Distributions in the Complex Wavelet Domain*, IEEE Signal Processing Letters, 12 (1), 2005, 17-20.

[26] Avindra F., *Image Denoising with the Non-local Means Algorithm*, University of Kansas, December 2010.

[27] Buades A., Coll B., Morel J.-M., *A non-local algorithm for image denoising*, Computer Vision and Pattern Recognition, 2005.

[28] Du Y., Kirenko I., *Benchmarking of Coding Artifacts Reduction Techniques*, Philips Restricted Technical Note PR-TN 2007/00681, October 2007.

[29] Wu H.R., Yuen M., *A generalized Block-edge impairment metric for video coding*, IEEE Signal Processing Letters, 4, November 1997, 317-320.