ADRIAN NAKONECHNYY, ZENOVIY VERES[*]

ADAPTIVE TRAINED FILTERS WITH SSIM OPTIMIZATION FOR VIDEO UPSCALING WITH SOFT PIXEL VALUE TRUNCATION

ADAPTACYJNE FILTRY UCZĄCE SIĘ Z OPTYMALIZACJĄ SSIM DLA ZWIĘKSZENIA ROZDZIELCZOŚCI OBRAZU Z MIĘKKIM PROGOWANIEM WARTOŚCI PIKSELI

Abstract

The paper presents the modified model for video upscaling based on the preservation of structural information from the input sequence in the adaptive trained filters. Additionally, it is proposed to make soft truncation of the result pixel value during the interpolation process. The evaluation of the proposed algorithm has shown good resulting for a variety of test sequences. The best results are obtained for a sequence with average bitrate and movement of the scene. Received results prove the algorithm is valuable for upscaling tasks.

Keywords: *image and video interpolation, up-scaling, image quality*

Streszczenie

W artykule przedstawiono zmodyfikowany model zwiększania rozdzielczości, oparty na przechowywaniu informacji strukturalnej z wejściowej sekwencji w adaptacyjnych filtrach uczących się. Dodatkowo zaproponowano miękkie progowanie wartości pikseli podczas procesu interpolacji. Weryfikacja proponowanego algorytmu wykazała korzystne rezultaty dla różnych sekwencji testowych. Najlepsze wyniki uzyskano w przypadku sekwencji o średniej szybkości transmisji i przy ruchomym obrazie. Wskazują one na korzyści wynikłe ze stosowania algorytmu w zadaniach zwiększania rozdzielczości obrazu.

Słowa kluczowe: *interpolacja zdjęć i video, zwiększanie rozdzielczości, jakość obrazu*

* Prof. D.Sc. Ph.D. Adrian Nakonechnyy, e-mail: adrnakon@polynet.lviv.ua, M.Sc. Zenoviy Veres, Department of Computerized Automatic Systems, Lviv Polytechnic National University.

1. Introduction

High definition television (HDTV) is becoming a standard appliance in every modern household. Also, with the introduction of HDTV-capable TV receivers, the transmission of standard definition television (SDTV) material will not stop immediately. In general, the price of high resolution screens has come down to a level that is affordable, even for TVs that have no HDTV reception. This raises the problem where low-resolution video materials have to be upscaled to fit the resolution of HDTV.

The set of linear techniques, such as nearest neighbor, bi-linear, and bi-cubic [8] interpolations have been popular in many applications. However, these linear methods usually result in blurred images, because the scaling process does not add new frequency components to interpolated images.

Several advanced non-linear image and video upscaling algorithms have been developed recently to deal with this problem [10]. Those video enhancement algorithms include sharpness enhancement, noise/coding artifacts reduction, resolution up-conversion, contrast enhancement, etc. This approaches provide good result image quality but requires much time in order to process the input image. This time consumption is acceptable for plain images, but they become a bottleneck for the video frames that are required to be transformed in the real time mode.

A family of trained filters [1] was developed that are used in real time mode for upscaling the video frames. The existing filter modifications uses different approaches in order to receive better output image quality [3]. In order to receive better quality of the output frames, in this paper, a novel interpolation technique based on optimizing filter coefficients based on structural information is presented. The proposed algorithm is evaluated using Mean Square Error (MSE) and Structural SIMilarity (SSIM) metrics.
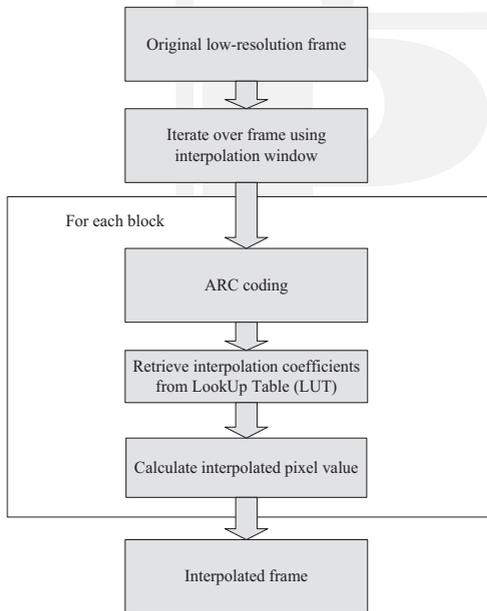
2. Trained Filters

Trained filters [1, 3, 6, 10] are widely used to solve image and video up-scaling tasks due to its simplicity. The interpolation process for trained filter is depicted at Fig. 1.

Low-resolution video frames are passed to the filter's input. Interpolation is performed in sliding window mode – one interpolated pixel value is evaluated based on all pixel values that belong to the interpolation window. The interpolation window represents a square block of 3×3 pixels. For each pixel to be interpolated, the pixels from the interpolation window

Fig. 1. Interpolation process with usage of trained filters

are classified using a classification method. The classification output provides the key for picking up filtering coefficients in the filter's look-up table (LUT) that are used for the pixel value calculation.

To obtain filtering coefficients of the filter the training process should be executed in advance. Fig. 2 depicts the training process of the trained filters. Original images are first downscaled by a factor of two according to the specification of the application. The training process employs the original video sequences and corresponding downscaled video sequences as the training material and uses the Least Mean Squares (LMS) criterion to get the optimal coefficients, which is computationally intensive due to the large number of classes. Fortunately, it needs to be performed only once.
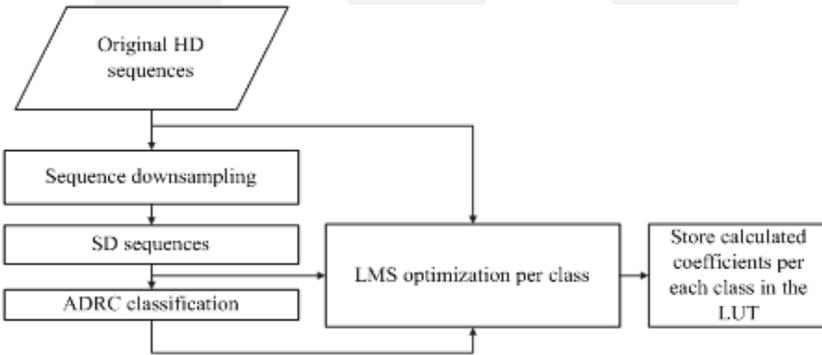

Fig. 2. Training process of the trained filter

Images after downsampling are referred to as downscaled images. In the downscaled images, each pixel and the pixels in its vicinity are characterized using a specific classification method. All the pixels and their neighbours belonging to a specific class and their corresponding pixels in the target (original) images are accumulated, and the optimal coefficients are obtained by making the MSE minimized statistically.

Let F_{SD}, F_{HD} represent pixels of the degraded images and the reference images for a particular class c, respectively. Then the interpolated pixel F_{HI} can be obtained by the desired optimal coefficients as follows:

$$F_{HI} = \sum_{k=0}^{n} w_c(k) \cdot F_{SD}(k) \qquad (1)$$

where:

$w_c(i), i \in [1 \dots n]$ – are the desired coefficients,
n – is the number of pixels in the aperture,
j – is the indicator of the particular aperture that represents the class c.

The summed square error between the filtered pixels and the reference pixels is:

$$e^2 = \sum_{j=1}^{N_c} \left(F_{R,c} - F_{F,c} \right)^2 = \sum_{j=1}^{N_c} \left[F_{R,c}(j) - \sum_{i=0}^{n} w_c(i) F_{D,c}(i,j) \right]^2 \qquad (2)$$

where:

N_c –   represents the number of training samples belonging to class c.

To minimize e^2, the first derivative of e^2 to $w_c(k)$, $k \in [1 \dots n]$ should be equal to zero.

$$\frac{\partial e^2}{\partial w_c(k)} = \sum_{j=1}^{N_c} 2 \cdot F_{D,c}(i,j) \cdot \left[F_{R,c}(j) - \sum_{i=0}^{n} \omega_c(i) F_{D,c}(i,j) \right] \qquad (3)$$

The optimal coefficients $w_c(k)$ are obtained by making the MSE minimized statistically. The calculated coefficients are then stored in a look-up table (LUT) for future use.

The local block content of the image can be classified based on the pattern of the image region and structure. Adaptive Dynamic Range Coding (ADRC) [1] is proposed as a powerful method for representing the structure of the region because of its high efficiency and simplicity. Let $x_1, x_2, \dots, x_n$ be pixel values, AV – is the average of all the pixel values in the aperture. The ADRC code per pixel is defined as follows:

$$\text{ADRC}(x_i) = \begin{cases} 1, & x_i > AV \\ 0, & \text{otherwise} \end{cases} \qquad (4)$$

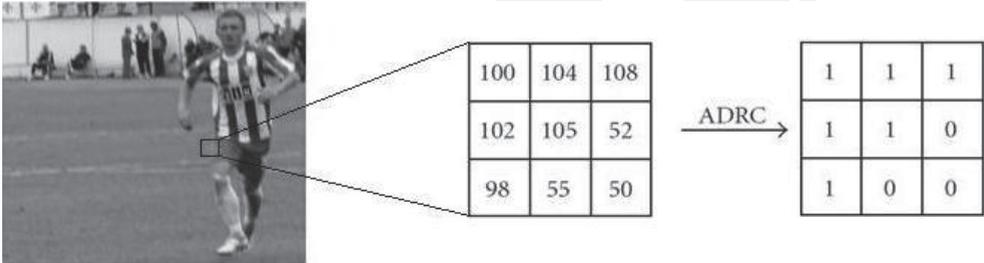ADRC coding diagram is shown in Fig. 3.

Fig. 3. ADRC coding of a 3 × 3 block

The main advantage of ADRC is its simple implementation. Classes count is decreased from 256^9 to 2^9 for apertures containing 3 × 3 SD pixels using Equation 1. It has been shown in [1] that if the image data is inverted, the coefficients in the LUT should remain the same.

By combining the two complementary classes, the size of the LUT is reduced to 2^8 without loss of image quality.

3. Main challenges of the trained filters

The optimal filtering coefficients are calculated during 'offline' training process. Filters incorporate the MSE metric for the coefficient's calculation that shows good results for the variety of tasks. However, this metric has several issues [7, 11]:

1. Digital pixel values, on which the calculation of this metric is based, may not accurately represent the light stimulation which is perceived by the human eye.
2. Human visual system's (HVS) sensitivity to errors is different for different types of errors and may also vary depending on image content. This difference is not counted correctly in the MSE calculation.
3. Two distorted images with the same errors energy may have different types of errors.
4. A simple summation of errors, which is implemented in the MSE calculation, is different from how the HVS and brain perceive the quality of the received image.

As a consequence, trained filters are sensitive to the same issues. The filter coefficients are calculated based on a predefined set of test sequences. The filters show good results for high-quality sequences if they were trained on high-quality ones but can fail when the sequence is distorted by noise. To address this issue, it is possible to create separate coefficients set for each input sequence type, but it will require much memory for their storage. To improve the quality of the result sequences, MSE metric is proposed to be replaced with another one. The structural similarity index – SSIM [4, 5] has been shown good results for objective evaluation of image and video quality. Usage of SSIM allows taking into account the structural information during coefficient's evaluation.

4. Structural Similarity Index [4]

The most fundamental principle underlying structural approaches to the image and video quality assessment (QA) is that the human visual system (HVS) is highly adapted to extract structural information from the visual scene. Therefore, a measurement of structural similarity or distortion should provide a good approximation to perceptual image quality.

The main idea that underlies the structural similarity (SSIM) index is a comparison of the distortion of three image components:

– Luminance
– Contrast
– Structure

Depending on how structural information and structural distortion are defined, there may be different ways to develop image QA algorithms. The SSIM index is a specific implementation from the perspective of image formation. The luminance of the surface of an object being observed is the product of illumination and the reflectance, but the structures of the objects in the scene are independent the illumination. Consequently, we wish to

separate the influence of illumination from the remaining information that represents object structures. Intuitively, the major impact of illumination change in the image is a variation of the average local luminance and contrast, and such variation should not have a strong effect on perceived image quality.

Consider two image patches, $\tilde{f}$ and $\tilde{g}$, obtained from the reference and test images. $\tilde{f}$ and $\tilde{g}$ consist of the two vectors of dimension N, where $\tilde{f}$ is composed of N elements of $f(n)$ spanned by a window B and similarly for $\tilde{g}$. To index each element of $\tilde{f}$, we use the notation $\tilde{f} = \left[\tilde{f}_1, \tilde{f}_2, \ldots, \tilde{f}_N \right]^{l}$.

First, the luminance of each signal is estimated as the mean intensity:

$$\mu_{\tilde{f}} = \frac{1}{N} \sum_{i=1}^{N} \tilde{f}_i, \quad \mu_{\tilde{g}} = \frac{1}{N} \sum_{i=1}^{N} \tilde{g}_i \tag{5}$$

A luminance comparison function $l(\tilde{f}, \tilde{g})$ is then defined as a function of $\mu_{\tilde{f}}$ and $\mu_{\tilde{g}}$:

$$l(\tilde{f}, \tilde{g}) = \frac{2\mu_{\tilde{f}}\mu_{\tilde{g}} + C_1}{\mu_{\tilde{f}}^2 + \mu_{\tilde{g}}^2 + C_1} \tag{6}$$

where:

C_1 – the constant that is included to avoid instability when $\mu_{\tilde{f}}^2 + \mu_{\tilde{g}}^2$ is very close to zero.

C_1 is taken as follows:

$$C_1 = (K_1 L)^2 \tag{7}$$

where:

L – is the dynamic range of the pixel values (255 for 8-bit grayscale images),
$K_1 \ll 1$ – is a small constant.

Similar considerations also apply to contrast comparison and structure comparison terms.

The contrast of each image patch is defined as an unbiased estimate of the standard deviation of the patch:

$$\sigma_{\tilde{f}}^2 = \frac{1}{N-1} \sum_{i=1}^{N} (\tilde{f}_i - \mu_{\tilde{f}})^2, \qquad \sigma_{\tilde{g}}^2 = \frac{1}{N-1} \sum_{i=1}^{N} (\tilde{g}_i - \mu_{\tilde{g}})^2 \tag{8}$$

The contrast comparison $c(\tilde{f}, \tilde{g})$ takes a similar form as the luminance comparison function and is defined as a function of $\sigma_{\tilde{f}}$ and $\sigma_{\tilde{g}}$:

$$c(\tilde{f}, \tilde{g}) = \frac{2\sigma_{\tilde{f}}\sigma_{\tilde{g}} + C_2}{\sigma_{\tilde{f}}^2 + \sigma_{\tilde{g}}^2 + C_2} \qquad (9)$$

where:

C_2 – is a nonnegative constant:

$$C_2 = (K_2 L)^2 \qquad (10)$$

where:

K_2 – satisfies $K_2 \ll 1$.

The signal is normalized (divided) by its own standard deviation so that the two signals being compared have unit standard deviation. The structure comparison $s(\tilde{f}, \tilde{g})$ is conducted on these normalized signals. The SSIM framework uses a geometric interpretation, and the structures of the two images are associated with the direction of the two unit vectors $\tilde{f} - \mu_{\tilde{f}}/\sigma_{\tilde{f}}$ and $\tilde{g} - \mu_{\tilde{g}}/\sigma_{\tilde{g}}$. The angle between the two vectors provides a simple and effective measure to quantify SSIM. In particular, the correlation coefficient between $\tilde{f}$ and $\tilde{g}$ corresponds to the cosine of the angle between them and is used as the structure comparison function:

$$s(\tilde{f}, \tilde{g}) = \frac{\sigma_{\tilde{f}\tilde{g}} + C_3}{\sigma_{\tilde{f}}\sigma_{\tilde{g}} + C_3} \qquad (11)$$

The covariance function $\sigma_{\tilde{f}\tilde{g}}$ between $\tilde{f}$ and $\tilde{g}$ is estimated as:

$$\sigma_{\tilde{f}\tilde{g}} = \frac{1}{N-1} \sum_{i=1}^{N} (\tilde{f}_i - \mu_{\tilde{f}})(\tilde{g}_i - \mu_{\tilde{g}}) \qquad (12)$$

Finally, the SSIM index between image patches $\tilde{f}$ and $\tilde{g}$ is defined as:

$$\text{SSIM}(\tilde{f}, \tilde{g}) = l(\tilde{f}, \tilde{g})^{\alpha} \cdot c(\tilde{f}, \tilde{g})^{\beta} \cdot s(\tilde{f}, \tilde{g})^{\gamma} \qquad (13)$$

where:

α, β and γ – are parameters used to adjust the relative importance of the three components.

In order to simplify the expression, in [2] these values were taken as $\alpha = \beta = \gamma = 1$ and $C_3 = C_2/2$. This results in a specific form of the SSIM index [2]:

$$\text{SSIM}(\tilde{f}, \tilde{g}) = \frac{(2\mu_{\tilde{f}}\mu_{\tilde{g}} + C_1)(2\sigma_{\tilde{f}\tilde{g}} + C_2)}{(\mu_{\tilde{f}}^2 + \mu_{\tilde{g}}^2 + C_1)(\sigma_{\tilde{f}}^2 + \sigma_{\tilde{g}}^2 + C_2)} \qquad (14)$$

Values of K_1 and K_2 in (3) are defined as $K_2 = 0.01$ and $K_2 = 0.03$.

The SSIM index and the three comparison functions – luminance, contrast, and structure – satisfy the following desirable properties:

1. Symmetry: $\mathrm{SSIM}(\tilde{f}, \tilde{g}) = \mathrm{SSIM}(\tilde{g}, \tilde{f})$. When quantifying the similarity between two signals, exchanging the order of the input signals should not affect the resulting measurement.

2. Boundedness: $\mathrm{SSIM}(\tilde{f}, \tilde{g}) \leq 1$. An upper bound can serve as an indication of how close the two signals are to being perfectly identical.

3. Unique maximum: $\mathrm{SSIM}(\tilde{f}, \tilde{g}) = 1$ if and only if $\tilde{f} = \tilde{g}$. The perfect score is achieved when the signals being compared are identical. In other words, the similarity measure should quantify any variations that may exist between the input signals.

The structure term of the SSIM index is independent of the luminance and contrast of the local patches, which is physically sensible because the change of luminance and/or contrast has little impact on the structures of the objects in the scene. Although the SSIM index is defined by three terms, the structure term in the SSIM index is regarded as the most important since variations in luminance and contrast of an image do not affect visual quality as much as structural distortions [7].

5. SSIM based trained filter

SSIM has been proven as a good alternative to MSE for objective image quality assessment [2, 11]. SSIM optimization will not change either the performance or complexity of the filtering process, but will be employed during the filter training process as depicted at Fig. 4.
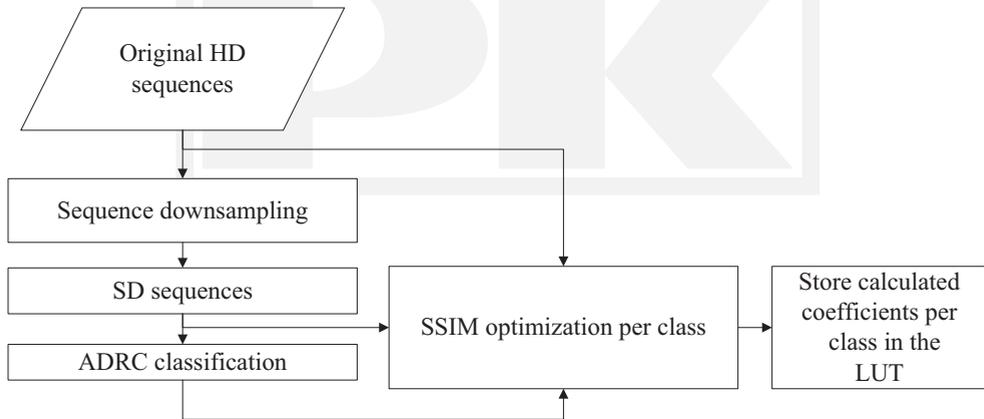

Fig. 4. Training process with SSIM optimization

Let F_{HD} – represents original HD data and F_{HI} – corresponding interpolated one from the set of SD pixels. The class c contains N samples during training. As a consequence (5), (6), (8), (12) are rewritten as:

$$\mu_{F_{\mathrm{HI}}} = \overline{F_{\mathrm{HI}}} = \frac{1}{N}\sum_{i=1}^{N}F_{\mathrm{HI}i}, \quad \mu_{F_{\mathrm{HD}}} = \overline{F_{\mathrm{HD}}} = \frac{1}{N}\sum_{i=1}^{N}F_{\mathrm{HD}i} \tag{15}$$

$$\sigma^2_{F_{\mathrm{HI}}} = \frac{1}{N-1}\sum_{i=1}^{N}(F_{\mathrm{HI}i} - \overline{F_{\mathrm{HI}}})^2, \quad \sigma^2_{F_{\mathrm{HD}}} = \frac{1}{N-1}\sum_{i=1}^{N}(F_{\mathrm{HD}i} - \overline{F_{\mathrm{HD}}})^2 \tag{16}$$

$$\sigma_{F_{\mathrm{HI}}F_{\mathrm{HD}}} = \frac{1}{N-1}\sum_{i=1}^{N}(F_{\mathrm{HI}i} - \overline{F_{\mathrm{HI}}})(F_{\mathrm{HD}i} - \overline{F_{\mathrm{HD}}}) \tag{17}$$

Finally, (14) becomes:

$$\mathrm{SSIM}_c = \frac{(2\mu_{F_{\mathrm{HI}}}\mu_{F_{\mathrm{HD}}} + C_1)(2\sigma_{F_{\mathrm{HI}}F_{\mathrm{HD}}} + C_2)}{(\mu^2_{F_{\mathrm{HI}}} + \mu^2_{\mathrm{FD}} + C_1)(\sigma^2_{F_{\mathrm{HI}}} + \sigma^2_{\mathrm{FD}} + C_2)} \tag{18}$$

Instead of SSIM calculation over apertures from original and interpolated frames, the similarity index is calculated and optimized based on pixel values that belong to specified class c over the all training data set.

To maximize SSIM_c, the first derivative of $\dfrac{\partial(\mathrm{SSIM}_c)}{\partial w_c(k)}$, $k \in [1 \ldots n]$ should be equal to zero:

$$\begin{cases} \dfrac{\partial(\mathrm{SSIM}_c)}{\partial w_c(1)} = 0 \\[2mm] \dfrac{\partial(\mathrm{SSIM}_c)}{\partial w_c(2)} = 0 \\[2mm] \ldots \\[2mm] \dfrac{\partial(\mathrm{SSIM}_c)}{\partial w_c(n)} = 0 \end{cases} \tag{19}$$

Equation (19) represents the system of non-linear equations since it contains squared values $\mu^2_{F_{\mathrm{HI}}}$ and $\sigma^2_{F_{\mathrm{HI}}}$. To figure out optimal $w_c(k)$ values one of the numerical methods should be employed. The Newton method for nonlinear systems is a proven solution [9] that has common usage for solving systems of nonlinear equations and provides a fast convergence.

6. Result pixel truncation

The usage of the SSIM faced with a problem of a pixel value overshot as described in [12]. In order to address this issue, it is proposed to introduce truncation into (1) during result pixel value calculation:

$$F_{\text{HI}}(i,j)_{\text{trunk}} = \begin{cases} F_{\text{SD MIN}}, & F_{\text{HI}}(i,j) \leq F_{\text{SD MIN}} \\ F_{\text{HI}}(i,j), & F_{\text{SD MIN}} < F_{\text{HI}}(i,j) < F_{\text{SD MAX}} \\ F_{\text{SD MAX}}, & F_{\text{SD MAX}} \geq F_{\text{SD MAX}} \end{cases} \qquad (20)$$

where:

$F_{\text{SD MIN}}$ and $F_{\text{SD MAX}}$ – the lowest and the highest pixel values from the interpolation window accordingly.

The experiments showed that the interpolated pixel value $F_{\text{HI}}(i,j)$ is never less than $F_{\text{SD MIN}}$.

Originally, the pixel truncation allows a small improvement in the image quality – less than 3% [12]. The proposed truncation is hard – the pixel overshot is not allowed at all. In order to receive better quality, it is proposed to make this truncation soft – 1% of pixel overshot is allowed; in case of the maximum pixel value in the block equals to the 240, the maximum allowed pixel value is 242. As a consequence, (20) is changed to the following:

$$F_{\text{HI}}(i,j)_{\text{trunk}} = \begin{cases} F_{\text{HI}}(i,j), & F_{\text{HI}}(i,j) < 1.01 \cdot F_{\text{SD MAX}} \\ F_{\text{SD MAX}}, & F_{\text{SD MAX}} \geq 1.01 \cdot F_{\text{SD MAX}} \end{cases} \qquad (21)$$

The original MSE-based interpolation filter requires the following memory to be allocated for the one calculation of the interpolated value within the 3×3 interpolation window:
– 512 bytes – for the LUT;
– 4 bytes – for the average of all the pixel values in the aperture (*AV* from (4));
– 9 bytes – to store pixel values ADRC code.

In total, the filter requires 525 bytes of memory except the one required to store input and output frames. The proposed algorithm requires two extra bytes of memory; it is less than 0.4% (527 vs. 525 bytes).

In order to receive one interpolated value, the original interpolation filter executes the following operations:
– 8 additions,
– 1 division,
– 9 comparisons,
– 9 multiplications.

The latency for the addition and comparison operations are equal for most count of the processors [13] and could be taken equal to 1. This parameter differs from processor to processor when the division operation is executed. The experiments were conducted on the Intel Core 2 processor (32 bit) with a latency value equal to 40 [13]. The result processor latency of the calculation equals 66. In order to fulfill (20), the filter additionaly requires 9 comparisons. As a consequence, the latency is increased by 13% – to 75. However, this change did not affect the processing time of the frame of the input sequence of HD format. This is explained by the processor's nature – the additional operations are executed in parallel. The majority of time is spent on the division operation.

7. Results

For objective evaluation, the MSE and SSIM between the original sequences and the result sequences processed on the down-scaled versions of the original sequences are calculated. The five sequences from the VQEG database were chosen for objective evaluation. The first four sequences have different compression levels, and the fifth is a raw sequence without compression. Table 1 shows the sequence characteristics.

T a b l e 1

Test sequences and their characteristics

Sequence	Characteristics	Bitrate
New York 2	movement	1.5 Mb/s
Mobile & Calendar	color, movement	768 Kb/s
Football	color, movement	3 Mb/s
Sailboat	almost still	4.5 Mb/s
Suzie	skin color	–

Table 2 shows the MSE results of upscaling between the original sequences (in raw format) and the upscaled ones. A trained filter performed the upscaling with the original coefficients (MSE optimized) and coefficients obtained using the proposed approach (SSIM optimized with truncation). The upscaled sequence was processed on the down sampled versions of the original sequences at different bitrates.

T a b l e 2

**MSE scores for resolution upscaling using
MSE and SSIM based ADRC**

Sequence	ADRC (MSE-based)	ADRC (SSIM-based)
NewYork 2	174.54	179.50
Mobile & Calendar	819.15	846.78
Football	1415.31	1414.33
Sailboat	191.23	234.43
Suzie	13.79	14.07

Table 3 shows the SSIM results evaluation. The results of the proposed algorithm are shown in the right column. All results are captured for the first 60 frames of the test sequences. The perfect SSIM score is achieved when the signals being compared are identical (SSIM = 1).

According to Table 2, the proposed algorithm has shown the worst results for MSE calculations, except in one case. This degradation is expected the since MSE-optimization approach was replaced by the SSIM based version.

Table 3

**SSIM scores for resolution upscaling using
MSE and SSIM based ADRC**

Sequence	ADRC (MSE-based)	ADRC (SSIM-based)
NewYork 2	0.912	0.924
Mobile & Calendar	0.293	0.293
Football	0.287	0.293
Sailboat	0.892	0.884
Suzie	0.946	0.958

The results of the SSIM metric calculation have shown better results for the proposed algorithm for three sequences (NewYork 2, Football, Suzie). Results are equals and worse in one case (Mobile & Calendar and Sailboat sequences respectively).

The SSIM and MSE results for the proposed algorithm are better for one sequence – Football. Usage of the SSIM optimization is beneficial when the filter is running against the sequence with average bitrates and movement of the scene. The usage of this approach over the sequences with small or no movement (Suzie and Sailboat) provides small benefits – the SSIM value is better for Suzie only while MSE values increased.

8. Conclusions

In this paper, the SSIM based coefficients calculation algorithm for the trained filter is presented. In the previous paper [12], it was proposed to introduce the truncation of the result pixel value. During the experiments, it was observed that the resulting pixel value is greater than the minimum pixel value from the interpolation window. Taking these observations into account allows for the elimination of the need to store the minimal pixel value in the memory. Additionally, it is proposed to make soft truncation and allow pixel overshot limited by 1% of the maximum pixel value.

The presented method has shown good results for various test video sequences. Since the simplest SSIM implementation was chosen for optimization and evaluation tasks, the observed results are similar for both algorithms. The received results allow for the focusing of further research on combining the SSIM and wavelet transformation to obtain better results for sequences with lower bitrates.

The proposed algorithm shows better results for the sequence with average bitrate and movement of the scene. This behavior is expected since the SSIM allows for focusing on the structural information from the scene rather than raw pixel values. The results prove that the algorithm is valuable for upscaling tasks and could be used in TV receivers as a replacement for the original MSE based version.

References

[1] Kondo T. et al., *Picture conversion apparatus, picture conversion method, learning apparatus and learning method*, US-patent 6, 323, 905, Nov. 2001.

[2] Zhao M. et al., *Content adaptive image de-blocking*, Proc. IEEE International Symposium on Consumer Electronics, Sep. 2004.

[3] Wang Z., Bovik A.C., Sheikh H.R., Simoncelli E.P., *Image Quality Assessment: From Error Visibility to Structural Similarity*, IEEE Transactions on Image Processing, Vol. 13, No. 4, Apr. 2004.

[4] Bovik A.C., *The Essential Guide to Image Processing*, Academic Press, 2009.

[5] Wang Z., Lu L., Bovik A.C., *Video quality assessment based on structural distortion measurement*, Signal processing: Image communication, Vol. 19, No. 2, Feb. 2004, 121-132.

[6] Shao L., An *Overview and Performance Evaluation of Least Mean Square Trained Filters*, Technical Note PR-TN 2007/00230 Philips Research Europe.

[7] Wang Z., Bovik A.C., *A universal image quality index*, IEEE Signal Process. Lett., 9(3), 2002, 81-84.

[8] Lehmann T.M., Gonner C., Spitzer K., *Survey: Interpolation Methods in Medical Image Processing*, IEEE Transactions On Medical Imaging, Vol. 18, No. 11, November 1999.

[9] Ortega M., Rheinboldt W.C., *Iterative Solution of Nonlinear Equations in Several Variables*, Classics in Applied Mathematics, SIAM, 2000.

[10] Veres Z., Nakonechnyy A., *Classifications of methods for video up-scaling*, Methods and instruments of quality control, No. 22, 2009, 76-80.

[11] Veres Z., Nakonechnyy A., *Image quality metrics and ways for their improvement*, Computer Technology In Printing, No. 20, 2008, 69-81.

[12] Veres Z., Nakonechnyy A., *Adaptive Trained Filters With SSIM Optimization and Pixel Value Truncation for Video Upscaling*, 7th International Conference Intelligent Data Acquisition and Advanced Computing Systems, Berlin, 12–14 September 2013, 404-407.

[13] Torbj̈orn Granlund, *Instruction latencies and throughput for AMD and Intel x86 processors*, (online) homepage: https://gmplib.org/~tege/x86-timing.pdf (date of access: 2014-04-27).