

Parameterization of the nonlinear interpolator invariant to four directions contours for multidimensional digital signals

A I Maksimov¹ and M V Gashnikov¹

¹Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086

Abstract. In this paper, a parameterization of the nonlinear interpolator invariant to four directions contours for the digital signals is described. The interpolator automatically selects one of the possible calculation methods for each signal sample based on the presence of the contour in a sample. Training procedure is performed to calculate the optimal value of the interpolator parameter. A criterion for minimizing the post-interpolation residues energy and a criterion for minimizing of the post-interpolation residues entropy are used to find the optimal value. The proposed interpolator is applied to the task of signal compression and the task of combining heterogeneous signals. In order to examine interpolators, computational experiments are carried out on a test set. The advantage in terms of the quadratic error of suggested interpolator over prototypes is shown.

1. Introduction

During recent years, the size of processed multidimensional digital signals has been expanding faster than the storage capacities. That is why compression methods nowadays are in a great demand.

Today, a large number of compression methods have been developed. JPEG method [1] is, perhaps, the most common of them. It uses statistical coding [2, 3] of transformants obtained by the two-dimensional discrete cosine transform [4]. Other popular compression methods are based on two-dimensional orthogonal transformations [5], fractals [6] and wavelets [7].

The listed methods have one common flaw – they are resource-intensive, which makes it difficult to use them in real-time systems, for example, in on-board imaging systems of satellites, drones and other aircrafts. These systems need compression methods that have little computational complexity. Methods which perform signal processing in a spatial domain, without transitioning to spectral components will be perfect for such application.

Methods based on differential pulse-code modulation (DPCM) [7-10] satisfy these requirements. Decorrelation of the signal in DPCM compression is achieved via the use of difference representation. However, the scope of DPCM is not limited to real-time systems, for example, DPCM acts as a stage in other compression methods. For example, JPEG algorithm uses DPCM compression at one of the stages of operation. Thus, the task of enhancement of DPCM and increasing its efficiency is topical. In this paper, we propose a new parameterized interpolator that makes it possible to increase the efficiency of DPCM compression by switching between different interpolation methods for each processed sample, depending on the presence of a contour in the vicinity of it.

The work is arranged in the following way. First, a general differential pulse-code modulation method, main types of DPCM interpolators and quantizers are presented. The next paragraph outlines

the interpolation and training procedures of proposed interpolator. The third paragraph shows the results of experimental study of proposed interpolator.

2. Differential pulse-code modulation

DPCM compression is the sequence of following procedures – interpolation, quantization, reconstruction, and coding. The generalization of this method is presented below.

2.1. General DPCM compression

Let $C(\bar{v})$ be the original (compressible) multidimensional digital signal where \bar{v} is the vector of its arguments. Differential pulse-code modulation can be represented as the following sequential procedure:

1. First, a reference value $\bar{C}(\bar{v})$ is interpolated;
2. Then, the differential signal $f(\bar{v})$ is calculated as the difference between reference and original value of the signal (1):

$$f(\bar{v}) = C(\bar{v}) - \bar{C}(\bar{v}), \quad (1)$$

where $C(n, m)$ – is the original value of the signal.

3. The resulting differential signal is quantized (2):

$$f_q(\bar{v}) = Q(f(\bar{v})), \quad (2)$$

where Q – is the quantization function.

4. The reconstructed signal value $\hat{C}(\bar{v})$ is calculated as the sum of the quantized differential signal and the reference value. The reconstructed value (3) is used to further interpolation:

$$\hat{C}(\bar{v}) = \bar{C}(\bar{v}) + f_q(\bar{v}); \quad (3)$$

5. After processing the entire signal, the quantized differential signal $f_q(n, m)$ is statistically encoded.

2.2. DPCM compression: Interpolation

High DPCM compression ratio is achieved when the difference signal is close to zero. Thus, it is necessary to interpolate the reference value as close to the reference one as possible. There are three main types of DPCM interpolators: averaging [8], nonlinear [9] and adaptive [10].

Averaging (linear) interpolators calculate the reference value the average over several previous signal samples. The simplest linear interpolator is the one that chooses the previous sample as the reference value. Let us describe the work of interpolators for the case of a two-dimensional original (compressible) signal $C(\bar{v}) = C(n, m)$.

Figure 1 shows the work of the interpolator, averaging over the two previous samples [8]. The interpolated value in this case is formed as follows:

$$\bar{C}(n, m) = \left[\frac{\hat{C}(n-1, m) + \hat{C}(n-1, m-1) + \hat{C}(n+1, m-1)}{4} \right]. \quad (4)$$

Interpolators of this type are robust to noise, since they use averaging over several samples, but they can have a huge error on the contours.

Non-linear interpolators have a small error on the contours. The main idea of such interpolators is to interpolate the value of the signal "along" the contour using the values of the previous signal samples.

Graham interpolator [9] is a non-linear one. It is able to interpolate "along" horizontal and vertical contours. As the reference value the sample corresponding to the smallest of the following differences (5,6,7) is taken:

$$\bar{C} = \begin{cases} \hat{C}(n, m-1), & \text{if } \lambda_n > \lambda_m; \\ \hat{C}(n-1, m), & \text{if } \lambda_n \leq \lambda_m, \end{cases} \quad (5)$$

here

$$\lambda_n = |\hat{C}(n, m-1) - \hat{C}(n-1, m-1)|, \quad (6)$$

$$\lambda_m = |\hat{C}(n-1, m) - \hat{C}(n-1, m-1)|, \quad (7)$$

where λ_n – is the horizontal difference; λ_m – is the vertical difference. Figure 2 shows, how this interpolator works.

The interpolator invariant to the four directions contours [10] is the enhancement of Graham interpolator. It is able to interpolator "along" horizontal, vertical and two diagonal contours. To select the reference value, 12 differences between the 8 previous samples are calculated. The interpolation is performed as follows :

$$\bar{C} = \begin{cases} \hat{C}(n, m-1), & \text{if } \lambda_{\downarrow} = \min(\lambda); \\ \hat{C}(n-1, m), & \text{if } \lambda_{\leftarrow} = \min(\lambda); \\ \hat{C}(n-1, m-1), & \text{if } \lambda_{\swarrow} = \min(\lambda); \\ \hat{C}(n+1, m-1), & \text{if } \lambda_{\searrow} = \min(\lambda), \end{cases} \quad (8)$$

here

$$\lambda_{\downarrow} = |\hat{C}(n, m-1) - \hat{C}(n-1, m-1)| + |\hat{C}(n-1, m-1) - \hat{C}(n-2, m-1)| + |\hat{C}(n-1, m+1) - \hat{C}(n-2, m+1)|, \quad (9)$$

$$\lambda_{\leftarrow} = |\hat{C}(n-1, m-1) - \hat{C}(n-1, m-2)| + |\hat{C}(n-1, m) - \hat{C}(n-1, m-1)| + |\hat{C}(n-1, m+1) - \hat{C}(n-1, m)|, \quad (10)$$

$$\lambda_{\swarrow} = |\hat{C}(n, m-1) - \hat{C}(n-1, m)| + |\hat{C}(n-1, m) - \hat{C}(n-2, m-1)| + |\hat{C}(n-1, m-1) - \hat{C}(n-2, m)|, \quad (11)$$

$$\lambda_{\searrow} = |\hat{C}(n, m-1) - \hat{C}(n-1, m-2)| + |\hat{C}(n-1, m) - \hat{C}(n-2, m-1)| + |\hat{C}(n-1, m+1) - \hat{C}(n-2, m)|, \quad (12)$$

where λ_{\downarrow} – is the vertical difference; λ_{\leftarrow} – is the horizontal difference; λ_{\swarrow} – is the 45° angle diagonal difference; λ_{\searrow} – is the 135° angle diagonal difference.

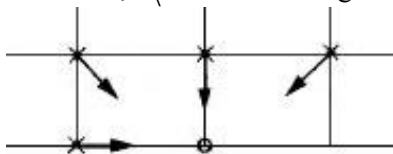


Figure 1. Averaging over four previous samples interpolator.



Figure 2. Graham interpolator.

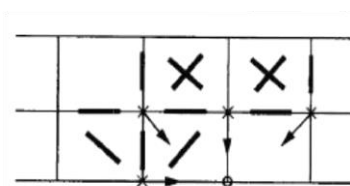


Figure 3. Interpolator invariant to the four directions contours.

Since nonlinear interpolators do not use averaging, they are subject to noise. Adaptive interpolators use the advantages of both averaging and non-linear interpolators. The adaptive interpolator at each point switches between the averaging and nonlinear method of interpolation of the reference value depending on the local features of the signal. To select an interpolation method, a criterion is chosen whose value at each point determines the choice of the interpolation method. Each interpolation method is meant to be used when it produces the smallest interpolation error - the averaging interpolator on the "smooth" sections of the signal, the nonlinear one - on the contours. The work of this interpolator is shown in Figure 3.

Adaptive Graham interpolator [10] is an example of adaptive interpolators. In each processed sample, the difference of the differences from formulas (6) and (7) is calculated. The calculated criterion is compared with the thresholds that are set for each processed signal by the preliminary training procedure. The interpolation procedure can be written as follows :

$$\bar{C} = \begin{cases} \hat{C}(n, m-1), & \text{if } \lambda < \lambda_{-}; \\ \hat{C}(n, m-1) + \hat{C}(n-1, m), & \text{if } \lambda \in [\lambda_{-}, \lambda_{+}]; \\ \hat{C}(n-1, m), & \text{if } \lambda > \lambda_{+}, \end{cases} \quad (13)$$

$$-C_{max} \leq \lambda_{-} \leq 0 \leq \lambda_{+} \leq C_{max}, \quad (14)$$

here

$$\lambda_n = |\hat{C}(n, m-1) - \hat{C}(n-1, m-1)|, \quad (15)$$

$$\lambda_m = |\hat{C}(n-1, m) - \hat{C}(n-1, m-1)|, \quad (16)$$

$$\lambda = \lambda_n - \lambda_m, \quad (17)$$

where λ is the contour criterion; λ_- is the lower threshold for switching between interpolation methods; λ_+ is the upper threshold for switching between interpolation methods; C_{max} is the maximum value of the signal. Figure 4 shows the work of adaptive Graham interpolator.

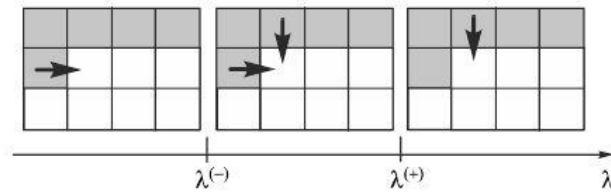


Figure 4. Adaptive Graham interpolator.

The disadvantage of this interpolation method is the need for a preliminary training procedure to calculate the thresholds which allow switching between interpolation methods.

2.3. DPCM compression: Quantization

Quantization [8,11] is the process of dividing the signal value range into a finite number of ranges, each of which is represented by one specific value.

Quantization allows increasing the compression ratio of the processed signal by introducing an error in it. The quantization of the difference signal has a special feature - a small number of quantization levels, therefore, they should be chosen thoroughly. In the present study, two quantization scales are used - uniform [12] and Max [13] quantization scales.

All intervals of the uniform quantization scale are equal, its representative values are located in the middle of the intervals and at equal distances from each other. The uniform quantization scale is a scale with a controllable error [14]. The maximum error introduced into the quantized signal is determined by the width of the scale interval. Uniform scale quantization can be written as follows:

$$f_q(\bar{v}) = \text{sign}(f(\bar{v})) \times \left[\frac{|f(\bar{v})| \times E_{max}}{2E_{max} + 1} \right], \quad (18)$$

where E_{max} – is the maximum error.

The advantage this quantization scale is the level proportionality of the quantized signal and the original one.

Intervals of Max quantization scale are not equal. Its levels are constructed on the basis of the mean-square quantization error minimization criterion [13]. With a known distribution density of the difference signal, expression (19) for the standard-mean quantization error can be written as follows:

$$\mathcal{E}_q^2 = \sum_{l=0}^{L-1} \int_{d_l}^{d_{l+1}} (C - r_l) p(C) dC, \quad (19)$$

where \mathcal{E}_q^2 – is the quantization root-mean-square error; L – is the number of quantization levels; d_l – is the border value; r_l – is the representative value; $p(C)$ – is the differential signal distribution density.

Since the difference signal is discrete, (19) can be written as:

$$\mathcal{E}_q^2 = \sum_{l=0}^{L-1} \sum_{C=d_l}^{d_{l+1}} (C - r_l) p(C). \quad (20)$$

To construct the Max scale, it is necessary to choose d_l and r_l which would minimize the error (20). Expressions (21,22) for them are found by taking the partial derivatives of expression (20):

$$r_i = \frac{\sum_{C=d_i}^{d_{i+1}} C p(C)}{\sum_{C=d_i}^{d_{i+1}} p(C)}, \quad (21)$$

$$d_j = \frac{r_{j-1} + r_j}{2}. \quad (22)$$

In practice, Max scale is constructed from the uniform one iteratively. Its border values are calculated with the use of its representative values, and vice versa. The resulting Max scale levels are more frequent on the signal value area where the values most often appear.

3. Parameterization of the nonlinear interpolator invariant to four directions contours

We propose a new adaptive interpolator. It is a parameterization of the interpolator invariant to the four directions contours. Interpolation and trainig procedures are developed for proposed interpolator.

During training procedure a preliminary pass through the original signal is required to calculate the threshold value, upon which the interpolation method is chosen.

The contour criterion for the given parameterized interpolator is the difference module between the reference values obtained by means of the interpolator averaging over the four previous readings (4) and the interpolator invariant to the four directions contours (8). If the difference absolute value between the averaging and nonlinear methods is small, averaging interpolation is chosen, since it is noise-resistant. If the difference absolute value is large, then a contour is detected and nonlinear interpolation is chosen. Let us describe the work of interpolator for the case of a two-dimensional original signal $C(\vec{v}) = C(n, m)$.

3.1. Interpolation procedure

The interpolation procedure for proposed interpolator can be represented as follows:

$$\begin{aligned} & (\hat{C}(n-1, m) + \hat{C}(n, m-1)\hat{C}(n-1, m-1) + \hat{C}(n+1, m-1))/4, \\ & \text{if } |\min(\lambda) - \lambda_m| \leq \Lambda; \\ \bar{C} = & \begin{cases} \hat{C}(n, m-1), & \text{if } \lambda_1 = \min(\lambda) \wedge |\lambda_1 - \lambda_m| > \Lambda; \\ \hat{C}(n-1, m), & \text{if } \lambda_- = \min(\lambda) \wedge |\lambda_- - \lambda_m| > \Lambda; \\ \hat{C}(n-1, m-1), & \text{if } \lambda_7 = \min(\lambda) \wedge |\lambda_7 - \lambda_m| > \Lambda; \\ \hat{C}(n+1, m-1), & \text{if } \lambda_\backslash = \min(\lambda) \wedge |\lambda_\backslash - \lambda_m| > \Lambda, \end{cases} \end{aligned} \quad (23)$$

here

$$\lambda_1 = |\hat{C}(n, m-1) - \hat{C}(n-1, m-1)| + |\hat{C}(n-1, m-1) - \hat{C}(n-2, m-1)| + |\hat{C}(n-1, m+1) - \hat{C}(n-2, m+1)|, \quad (24)$$

$$\lambda_- = |\hat{C}(n-1, m-1) - \hat{C}(n-1, m-2)| + |\hat{C}(n-1, m) - \hat{C}(n-1, m-1)| + |\hat{C}(n-1, m+1) - \hat{C}(n-1, m)|, \quad (25)$$

$$\lambda_7 = |\hat{C}(n, m-1) - \hat{C}(n-1, m)| + |\hat{C}(n-1, m) - \hat{C}(n-2, m-1)| + |\hat{C}(n-1, m-1) - \hat{C}(n-2, m)|, \quad (26)$$

$$\lambda_\backslash = |\hat{C}(n, m-1) - \hat{C}(n-1, m-2)| + |\hat{C}(n-1, m) - \hat{C}(n-2, m-1)| + |\hat{C}(n-1, m+1) - \hat{C}(n-2, m)|, \quad (27)$$

$$\lambda_m = \left[\frac{\hat{C}(n-1, m) + \hat{C}(n-1, m) + \hat{C}(n-1, m-1) + \hat{C}(n+1, m-1)}{4} \right], \quad (28)$$

where Λ – is the interpolation threshold. Figure 5 shows the work of proposed interpolator. The procedure for calculation of the threshold value Λ is described in the next section of the article.

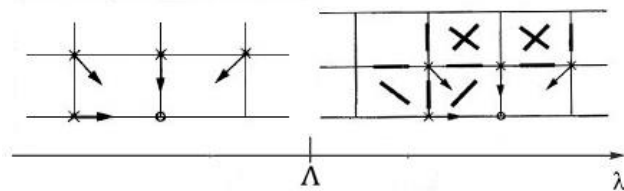


Figure 5. Parameterized nonlinear interpolator invariant to four directions contours.

3.2. Post-interpolation residues energy minimization-based training procedure

To determine the value of the threshold Λ used for switching interpolation methods, it is necessary to solve the following optimization problem (29):

$$\delta(\Lambda) = \sum_{(m,n) \in \omega} |C(n, m) - \hat{C}(n, m)| \rightarrow \min_{\Lambda}. \quad (29)$$

where δ – is the interpolation error (the energy of post-interpolation residues); ω – is the set of signal samples coordinates.

To solve this problem, a preliminary signal pass is required during which the matrix Δ (30-32) is filled. Its elements contain the total interpolation error (the sum of the difference modules between the signal sample and the interpolated value) for which criterion equals λ :

$$\Delta(1, \lambda) = \sum_{m, n \in w(\lambda)} |C(n, m) - \min_{\lambda} \{\hat{C}(n-1, m), \hat{C}(n, m-1), \hat{C}(n-1, m-1), \hat{C}(n+1, m-1)\}|, \quad (30)$$

$$\Delta(2, \lambda) = \sum_{m, n \in w(\lambda)} |C(n, m) - \left\lfloor \frac{\hat{C}(n-1, m) + \hat{C}(n, m-1) + \hat{C}(n-1, m-1) + \hat{C}(n+1, m-1)}{4} \right\rfloor|, \quad (31)$$

$$-C_{max} \leq \lambda \leq C_{max}, \quad (32)$$

where $w(\lambda)$ – is the set of λ values. The error vector is filled via recursive procedure:

$$\delta(\lambda_{max}) = \sum_{\lambda=0}^{C_{max}} \Delta(1, \lambda), \quad (33)$$

$$\delta(\lambda) = \delta(\lambda + 1) - \Delta(1, \lambda) + \Delta(2, \lambda), C_{max} - 1 \geq \lambda \geq 0. \quad (34)$$

After this operation, the value of the threshold Λ can be found from expression (29). This vector contains $C_{max} + 1$ elements so one can find the required threshold value by a simple search. Thus, the optimization problem (29) is solved.

It should be noted that the proposed interpolator, including the described optimization procedure, can be used not only for compression, but also for other signal processing tasks, for example, for solving the problem of combining heterogeneous signals which have different resolution, number of components, etc.

3.3. Post-interpolation residues entropy minimization-based training procedure

When solving the compression problem, instead of (29), it is also worth considering a criterion for minimizing the amount of compressed data. In this paper we consider the quantized post-interpolation residues entropy of (18) as an estimation of the amount of compressed data:

$$H(\Lambda) = - \sum_{q=-C_{max}}^{C_{max}} N_q(\Lambda) \ln N_q(\Lambda) \rightarrow \min_{\Lambda} \quad (35)$$

where $N_q(\Lambda)$ is the amount of the quantized post-interpolation residuals with a value q .

To solve the optimization problem (35), a three-dimensional matrix is used:

$$\mathbf{N}_{\lambda, q}^{(i)}, i \in \{1, 2\}, 0 \leq \lambda < C_{max} \quad (36)$$

Each element of $\mathbf{N}_{\lambda, q}^{(i)}$ contains the amount of quantized post-interpolation residues (18) with a value q , given the feature value (24) λ and interpolator number i , which is taken from (23). $i=0$ means that the first row of (23) is taken as the interpolator, for $i=1$ all other rows of (23) are taken as interpolator.

Matrix $\mathbf{N}_{\lambda, q}^{(i)}$ is used to calculate the amount of quantized post-interpolation residuals (18) $N_q(\lambda)$ with value q for all possible values of the threshold λ :

$$N_q(0) = \sum_{\lambda=0}^{C_{max}-1} \mathbf{N}_{\lambda, q}^{(1)}, N_q(\lambda+1, q) = N_q(\lambda, q) - \mathbf{N}_{\lambda, q}^{(1)} + \mathbf{N}_{\lambda, q}^{(2)}. \quad (37)$$

The number of values of the quantized post-interpolation residues (18) equal to q makes it possible to calculate a one-dimensional array of entropy values (35) for all possible values of the threshold λ . The length of this array is small (C_{max} of the elements, i.e. equal to the maximum value of the signal). The number of the minimum entropy value (35) can be found by the direct search:

$$\Lambda = \arg \min_{\lambda} (H(\lambda)) \quad (38)$$

Thus, the optimization problem (35) is solved.

4. Experimental study of DPCM interpolators

For the developed interpolator based on criterion (29) a series of computational experiments was carried out, during which the dependencies of the root-mean-square error (RMS) on the compression ratio of the quantization scale were obtained. The uniform (18) and the Max scale (21, 22) were used. The mathematical expression for the root-mean-square is as follows:

$$\mathcal{E} = \frac{1}{\delta_C} \sum_{\bar{v} \in V} \sqrt{(C(\bar{v}) - \tilde{C}(\bar{v}))^2}, \quad (39)$$

where δ_C – is the original signal variance; V – is the set of signal samples; $C(\bar{v})$ – is the original signal values; $\tilde{C}(\bar{v})$ – is the processed (decompressed) signal values.

The results obtained for the developed interpolator were compared with the others. For comparison, prototypes on the basis of which a parameterized interpolator was developed were taken. They are averaging over the four previous samples (4) and invariant to the contours of the four directions (8-12) interpolators. As a test signal set Waterloo Grayscale Set 1 [15] was used. Figure 6 shows the test set. The computational experiment was carried out in the following way: the test signals were DPCM-compressed [16] and decompressed with a constant compression ratio of a uniform quantization scale and a selected interpolator, after which the error was calculated. The results for the selected interpolator were averaged over all test set. Then the procedure was repeated for the next compression ratio of the scale. Thus, the dependencies of the root-mean-square error on the compression ratio of a uniform quantization scale were constructed for all three interpolators. A similar series of experiments was carried out for the Max quantization scale. Figures 7 and 8 show the results of the computational experiments.

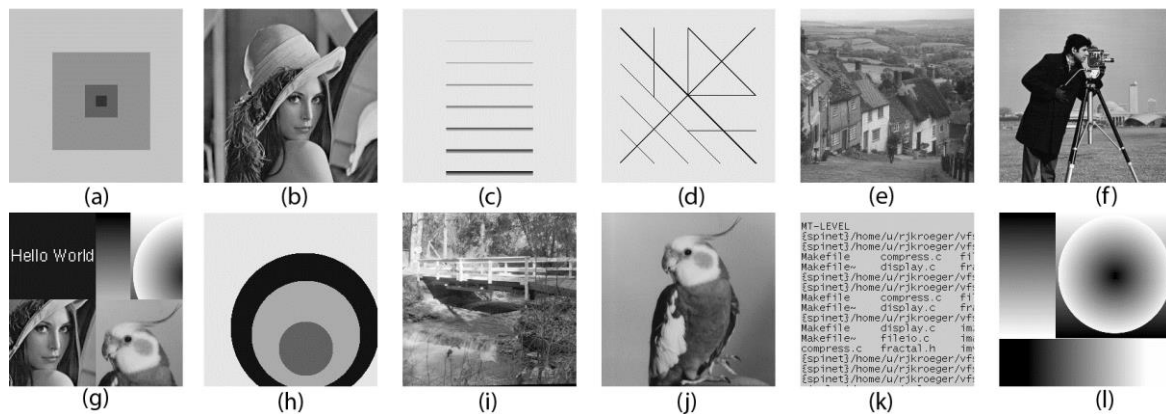


Figure 6. Waterloo Grayscale Set 1 test signals, signals (a), (c), (d), (h) are inverted.

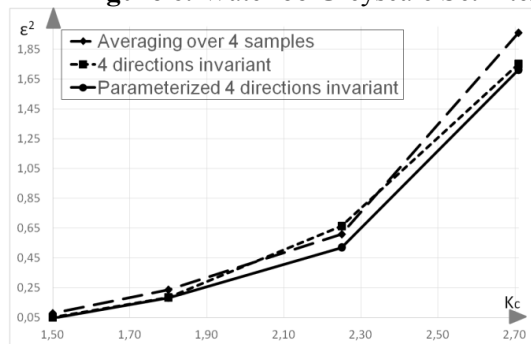


Figure 7. Dependency of RMS on compression ratio for uniform quantization scale for each interpolator.

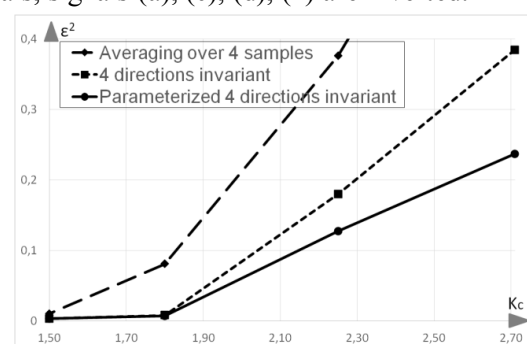


Figure 8. Dependency of RMS on compression ratio for Max quantization scale for each interpolator.

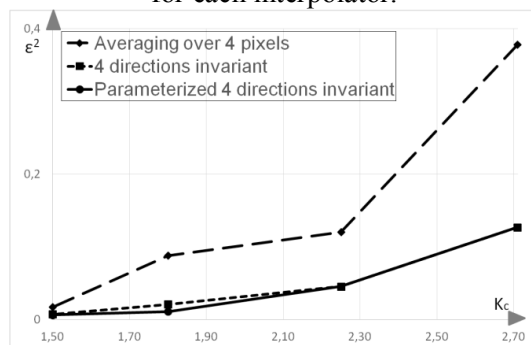


Figure 9. Dependency of RMS on compression ratio for uniform quantization scale for Fig.6c.

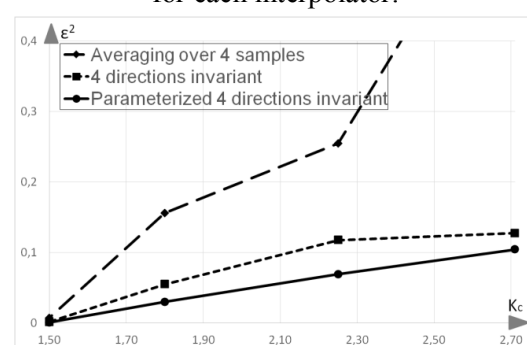


Figure 10. Dependency of RMS on compression ratio for Max quantization scale for Fig.6c.

As can be seen from the presented dependencies, the developed parameterization outperforms its prototypes. During computational experiments, the type of signal was found, with which the developed parameterization gives the least RMS. This result is shown in Figures 9 and 10.

5. Conclusions

This article presents a new adaptive interpolator for DPCM-signal compression, based on the parameterization of the interpolator, which is invariant to four directions contours. Its interpolation and training procedures are presented. Training procedure includes a preliminary pass through the signal.

A series of computational experiments was carried out. The developed parameterization outperformed its prototypes. During the computational experiments, the type of signals on which the proposed interpolator shows the best results was determined.

6. References

- [1] Gupta V 2014 Enhanced Image Compression Using *Wavelets International Journal of Research in Engineering and Science (IJRES)* **2(5)** 55-62
- [2] Sayood K 2012 *Introduction to Data Compression* (The Morgan Kaufmann Series in Multimedia Information and Systems)
- [3] Wallace G 1991 The JPEG Still Picture Compression Standard *Communications of the ACM*, **34(4)** 30-44
- [4] Huffman D A 1952 A Method for the Construction of Minimum Redundancy Codes *Proc. IRE*, **40** 1098-1101
- [5] MacKay D 2003 *Information Theory, Inference, and Learning Algorithms* (Cambridge Univ. Press)
- [6] Plonka G M 2005 Fast and numerically stable algorithms for discrete cosine transforms *Linear Algebra and its Applications* **394(1)** 309-345
- [7] Woon W M 2000 Achieving high data compression of self-similar satellite images using fractal *Proceedings of IEEE International Geoscience and Remote Sensing Symposium* 609-611
- [8] Pratt W 2007 *Digital Image Processing* (Wiley)
- [9] Gonzalez R 2008 *Digital Image Processing* (Pearson Education)
- [10] Soifer V A 2010 *Computer Image Processing. Part II: Methods and algorithms* (VDM Verlag)
- [11] Gashnikov M V 2017 Parameterized adaptive predictor for digital image compression based on the differential pulse code modulation *Proceedings of SPIE The International Society for Optical Engineering* 1034110 DOI: 10.1117/12.2268530
- [12] Sayood K 2006 *Introduction to data compression* (Morgan Kaufmann Publishers)
- [13] Salomon D 2007 *Data Compression. The Complete Reference* (Springer-Verlag)
- [14] Lin S 2004 *Error Control Coding: Fundamentals and Applications, second edition* (New Jersey: Prentice-Hall, inc)
- [15] *Image Repository: The Waterloo Fractal Coding and Analysis Group* (Access mode: <http://links.uwaterloo.ca/Repository.html>) (17.11.2017)
- [16] Gashnikov M V 2016 Parameterization of nonlinear Grahm predictor for digital image compression *Computer Optics* **40(2)** 225-231 DOI: 10.18287/2412-6179-2016-40-2-225-231

Acknowledgements

The reported study was funded by RFBR according to the research projects № 18-01-00667, № 18-07-01312.