Adaptation of parameterized interpolation algorithms of multidimensional signals for hierarchical and interpolation compression methods

Mikhail Gashnikov
Department of GIS and ITsec
Samara National Research University;
MMIP laboratory
Image Processing Systems Institute of RAS - Branch of the FSRC "Crystallography and Photonics" RAS
Samara, Russia
mih-fastt@yandex.ru

***Abstract*—We adapt parameterized multidimensional signal interpolators for hierarchical compression methods and interpolation compression methods based on the coding of quantized post-interpolation residues. The considered interpolators automatically select the most appropriate interpolating function at each point of the signal using a parameterized decision rule. We propose a set of interpolation functions for these compression methods. We select the optimization criterion for the proposed interpolator. The optimization criterion is based on minimization the entropy of the quantized post-interpolation residues. We solve the optimization problem of the adaptive parameterized interpolator according to this criterion. We perform computational experiments to study the proposed interpolators in natural multidimensional signals. The experimental results confirm that the use of the adaptive interpolators can significantly increase the effectiveness of the mentioned methods of multidimensional signals compression.**

Keywords—multidimensional signal, heterogeneous signals, adaptive interpolation, decision rule, interpolation error

I. Introduction

A large number of signal compression algorithms are known [1-10]: methods based on wavelet transform [2] (including JPEG-2000 [3]), fractal methods [4], DOP methods (including JPEG [5] based on DCT [6]), etc. This work deals with interpolation methods [7-8] and hierarchical methods [9-10] of multidimensional signals compression.

Interpolation compression methods [7-8], as their name implies, are based on the interpolation of signal samples from other (reference) samples of the same signal and the subsequent efficient coding [11-12] of post-interpolation residues.

Hierarchical compression methods [9-10] are based on a hierarchical (pyramidal) signal representation, which allows us to interpolate more down-sampled levels of the signal samples "pyramid" from less down-sampled levels. Then we encode for the errors of this interpolation.

The most important step in the last two compression methods is the interpolation algorithm. One of the most promising interpolators is the adaptive algorithm [13-14], which selects an interpolating function at each signal point using a parameterized decision rule. In this paper, we perform the adaptation of such algorithm for hierarchical methods and interpolation methods of multidimensional signals compression.

II. Adaptation of the Parametrized Algorithm for Interpolation Compression Methods

A. *Adaptive interpolator of multidimensional signal*

We interpolate a multidimensional signal sample $x(\vec{n})$ based on reference samples $\{\hat{x}_k(\vec{n})\}$. We select an interpolating function $U^{(i)}\left(\{\hat{x}_k(\vec{n})\}\right)$ for each sample with coordinates $\vec{n}$. We select the function $U^{(i)}$ for each sample using the parameterized rule *P*, depending on the vector parameter $\vec{t}$:

$$u(\vec{n}) = U^{(i)}\left(\left\{\hat{x}_k(\vec{n})\right\}\right), \quad i = P\left(\vec{f}(\vec{n}), \vec{t}\right), \tag{1}$$

where $u(\vec{n})$ is the interpolating value, $\vec{t}$ is the vector of parameters, $\vec{f}(\vec{n})$ is the vector of local features. We calculate these local features based on the same reference samples $\{\hat{x}_k(\vec{n})\}$.

B. *Interpolation compression methods*

Interpolation compression methods work as follows. We select a reference samples set $\{\hat{x}_k(\vec{n})\}$ from the set $\{x(\vec{n})\}$ of all signal samples. We interpolate the remaining (intermediate) samples of the signal based on the reference samples:

$$u(\vec{n}) = U\left(\left\{\hat{x}_k(\vec{n})\right\}\right). \tag{2}$$

Then we calculate the difference signal (post-interpolation residuals):

$$\lambda(\vec{n}) = x(\vec{n}) - u(\vec{n}). \tag{3}$$

Then we quantize the difference signal using a quantization function Q:

$$q(\vec{n}) = Q(\lambda(\vec{n})). \quad (4)$$

Then we encode the post-interpolation residues $q(\vec{n})$ with some statistical encoding algorithm and put the encoded signal into the communication channel or archive file.

To adapt the parameterized interpolator to interpolation methods of signal compression, it is necessary to specify the following elements of this interpolator:

1) optimization criterion;

2) local features and decision rule;

3) optimization procedure for the parametrized interpolator;

4) interpolating functions.

C. *Adaptive interpolator optimization criterion for compression*

The criteria [11-12] based on the interpolation error minimization are usually used to optimize the interpolators. However, this work deals with a criterion more suitable for the compression problem. This criterion is based on minimization of the compressed data size. We use the unnormalized entropy of the quantized post-interpolation residuals as an estimate of this compressed data size:

$$H(\vec{t}) = -\sum_{q=-Q+1}^{Q-1} N_q(\vec{t}) \ln N_q(\vec{t}) \to \min_{\vec{t}}, \quad Q = \max_{\vec{n}}(q(\vec{n})) \quad (5)$$

where $N_q(\vec{t})$ is the number of quantized post-interpolation residues equal to q for a fixed parameter $\vec{t}$ of the adaptive interpolator.

D. *Decision rule and local features*

Averaging (smoothing) interpolator [7-8] is the simplest for interpolation compression:

$$u(\vec{n}) = \frac{1}{N}\sum_{k=1}^{N} \hat{x}_k(\vec{n}), \quad (6)$$

where N is the number of reference samples.

This interpolator is quite precise inside smoothly changing areas of the signal, since averaging filters noise a little. However, the error of the averaging interpolator almost always increases substantially at the boundaries of these smooth regions. However, the error of the averaging interpolator usually increases substantially at the boundaries of these smooth areas.

To interpolate these boundaries, nonlinear algorithms implementing interpolation "along" the boundaries are more efficient. For example, the Graham interpolator [12] works exactly in this way in the two-dimensional case. There are also modifications of this interpolator to the case of more than two directions [8]. However, nonlinear interpolators lose accuracy within smoothly varying areas of the signal.

We suggest using the adaptive interpolator described above for interpolation compression methods. This algorithm combines the advantages of averaging and non-linear approaches to interpolation. The adaptive interpolator can automatically select the "averaging" or "nonlinear" interpolating function depending on the direction and the severity of the boundary of smoothly changing regions in the neighborhood of each processed sample.

Denote by N_c the number of possible directions of the regions boundaries. Let $\{\delta_i(\vec{n}) : 0 \le i < N_c\}$ be the set of differences $\delta_i(\vec{n}) = \left|\hat{x}_\xi(\vec{n}) - \hat{x}_\varsigma(\vec{n})\right|$ between the reference samples $\hat{x}_\xi(\vec{n})$, $\hat{x}_\varsigma(\vec{n})$ in the considered directions. This set of differences $\delta_i(\vec{n})$ describes the severity (and the fact of presence) of the boundary of the regions in the neighborhood of the current reference with coordinates $\vec{n}$.

We detect the boundary and calculate its direction by means of a decision rule depending on the vector parameter $\vec{t}$. This parameter consists of several threshold values t_i. The decision rule compares the differences $\delta_i(\vec{n})$ with these thresholds. If there is no border at a current signal point, then the decision rule selects the "averaging" interpolating formula of the form (6):

$$u(\vec{n}) = U^{(1)}(\vec{n}) = \frac{1}{N}\sum_{k=1}^{N} \hat{x}_k(\vec{n}), \quad if\ \delta_i \le t_i,\ i \in [0, N_c) \quad (8)$$

If there is a boundary at the current point, then the decision rule selects the average value $\tilde{x}(\vec{n})$ of the nearest reference samples located "along" the boundary:

$$u(\vec{n}) = U^{(2)}(\vec{n}) = \tilde{x}_j(\vec{n}), \quad if\ \delta_i > t_i,\ i \in [0, N_c). \quad (9)$$

Therefore, we describe decision rule (1) by expressions (8-9). We need to solve the optimization problem (5) in the parametric space of t_i of dimension N_c to calculate the parameters t_i of this decision rule.

The compression task often imposes restrictions on computing resources, and the complexity of the interpolator optimization in the N_c-dimensional parametric space of the decision rule can become a source of problems even in the case of a three-dimensional or even two-dimensional signal.

We propose using a reduced-dimensional parametric space for the interpolator optimization as part of interpolation compression methods. In this case, the decision rule instead of differences δ_i uses relies on their ratio. We describe these relations by a variational series [8] $\delta^{(0)}(\vec{n}) \le \delta^{(1)}(\vec{n}) \le \delta^{(2)}(\vec{n}) \le .. \le \delta^{(N_c-1)}(\vec{n})$, in which $\delta^{(i)}$ are renumbered differences δ_i.

If there are no boundaries of smooth regions in the neighborhood of the current signal point, then all difference δ_i s have similar values. If the boundary is present, then the difference corresponding to the direction of this boundary is minimal. Besides, this difference is at the initial (zero) position $\delta^{(0)}$ of the variational series.

The remaining differences have similar meanings. The difference $\delta^{(0)}$ in this case differs significantly from the other differences $\delta^{(1)}..\delta^{(N_c-1)}$. Therefore, we can calculate the feature $f\left(\vec{n}\right)$ of the severity and direction of the boundary at the current signal point by means of the rank filter:

$$f\left(\vec{n}\right) = \delta^{(1)}\left(\vec{n}\right) - \delta^{(0)}\left(\vec{n}\right). \quad (11)$$

We use this characteristic as the feature of the decision rule (1) to select the interpolating function at each signal point.

If the feature $f\left(\vec{n}\right)$ is less than the threshold t, then there is no boundary at this signal point, i.e. you can use the "averaging" formula (8). Otherwise, we apply interpolation (9) “along” the boundary corresponding to the minimum difference δ_i:

$$u\left(\vec{n}\right) = \begin{cases} U^{(1)}\left(\vec{n}\right) = \sum_{k=1}^{N} \hat{x}_k\left(\vec{n}\right), & f\left(\vec{n}\right) \le t \\ U^{(2)}\left(\vec{n}\right) = \tilde{x}_j\left(\vec{n}\right), \ j = \arg\min_i \left\{\delta_i\left(\vec{n}\right)\right\}, & f\left(\vec{n}\right) > t \end{cases} \quad (12)$$

Thus, the described adaptive interpolator depends on the single scalar parameter t, and the problem of its optimization becomes one-dimensional.

E. *Optimization of adaptive interpolator*

We first fill out the three-dimensional auxiliary array $\mathbf{N}^{(i)}_{f',q'}$ of number of quantized post-interpolation residues (4) to optimize entropy (5):

$$\begin{gathered} \mathbf{N}^{(i)}_{f',q'} = \left|\left\{\vec{n} : f\left(\vec{n}\right) = f', q_i\left(\vec{n}\right) = q'\right\}\right|, \\ i \in \{1,2\},\ 0 \le f' < Q,\ -Q < q' < Q,\ Q = \max_{\vec{n}}\left(q\left(\vec{n}\right)\right) \end{gathered} \quad (13)$$

where

$$q_i\left(\vec{n}\right) = Q\left(x\left(\vec{n}\right) - U^{(i)}\left(\vec{n}\right)\right). \quad (14)$$

Each element $\mathbf{N}^{(i)}_{f',q'}$ contains the number of quantized post-interpolation residues $q_i\left(\vec{n}\right)$ (12), equal q', with the value of feature (11), equal f'.

We use the array $\mathbf{N}^{(i)}_{f,q}$ in the recursive procedure for calculating the number $N_q(t)$ of quantized post-interpolation residues (4) equal to q for all threshold values t:

$$N_q(0) = \sum_{f=0}^{M-1} \mathbf{N}^{(1)}_{f,q};\ N_q(t+1) = N_q(t) - \mathbf{N}^{(1)}_{t,q} + \mathbf{N}^{(2)}_{t,q} \quad (15)$$

The number of quantized post-interpolation residues $N_q(t)$ allows us to calculate the entropy $H(t)$ of quantized post-interpolation residues for all thresholds t

$$H(t) = -\sum_{q=-Q+1}^{Q-1} N_q(t)\ln N_q(t),\ Q = \max_{\vec{n}}\left(q\left(\vec{n}\right)\right) \quad (16)$$

The minimum value index $t = \arg\min_{t'} H(t')$ in the short array $H(t)$ is the solution to the optimization problem (5).

F. *Interpolation functions of the adaptive interpolator during compression*

We write the interpolating functions for the interpolation method of compression D-dimensional signal $x\left(\vec{n}\right) = x\left(n_0, ..., n_{D-1}\right)$. Let the samples $x\left(2\vec{n}\right)$ with even numbers be the reference. First, we specify the differences $\delta_i\left(2\vec{n}+1\right), i \in [0, D]$ between the reference samples:

$$\begin{gathered} \delta_i\left(2\vec{n}+1\right) = \Big| x\left(2\left(n_0+\eta_{i,0}\right), ..., 2\left(n_{D-1}+\eta_{i,D-1}\right)\right) \\ - x\left(2\left(n_0+1-\eta_{i,0}\right), ..., 2\left(n_{D-1}+1-\eta_{i,D-1}\right)\right)\Big|, i \in \left[0, 2^{D-1}\right) \end{gathered} \quad (17)$$

Here, the matrix $\eta_{i,d}$ defines all possible offsets of the reference samples relative to the interpolated sample, satisfying the condition $\sum_d \eta_{i,d} < 2^{D-1}$.

Next, we write the first interpolating function, averaging the neighborhood reference samples:

$$\begin{gathered} U^{(1)}\left(2\vec{n}+1\right) = \\ \frac{1}{2^{D-1}} \sum_{d=0}^{2^{D-1}-1} x\left(2\left(n_0+\eta_{i,0}\right), ..., 2\left(n_{D-1}+\eta_{i,D-1}\right)\right) + \\ \frac{1}{2^{D-1}} \sum_{d=0}^{2^{D-1}-1} x\left(2\left(n_0+1-\eta_{i,0}\right), ..., 2\left(n_{D-1}+1-\eta_{i,D-1}\right)\right) \end{gathered} \quad (18)$$

Then we write the averaging in the directions:

$$\begin{gathered} u_i^{(2)}\left(2\vec{n}+1\right) = \frac{1}{2}\Big(x\left(2\left(n_0+\eta_{i,0}\right), ..., 2\left(n_{D-1}+\eta_{i,D-1}\right)\right) \\ + x\left(2\left(n_0+1-\eta_{i,0}\right), ..., 2\left(n_{D-1}+1-\eta_{i,D-1}\right)\right)\Big), i \in \left[0, 2^{D-1}\right) \end{gathered} \quad (19)$$

Then we can write the adaptive interpolator (12), which selects one of the described interpolating functions at each point of the signal:

$$u\left(\vec{n}\right) = \begin{cases} U^{(1)}\left(\vec{n}\right), & f\left(\vec{n}\right) \le t \\ u_j^{(2)}\left(\vec{n}\right), \ j = \arg\min_i \left\{\delta_i\right\}, & f\left(\vec{n}\right) > t \end{cases} \quad (20)$$

III. Adaptation of the Parameterized Interpolator for Hierarchical Compression Methods

Hierarchical compression methods use a redundant pyramidal representation of a multidimensional signal $\mathbf{X} = \left\{x\left(\vec{n}\right)\right\}$ in the form of a set of L scale levels $\mathbf{X}_l$:

$$\mathbf{X} = \bigcup_{l=0}^{L-1} \mathbf{X}_l,\ \mathbf{X}_l = \left\{ \mathbf{X}_l(\vec{n}) \right\} = \left\{ \mathbf{X}(\vec{n}) : \vec{n} \in I_l \right\},$$
$$I_{L-1} = \left\{ 2^{L-1}\vec{n} \right\}, I_l = \left\{ 2^l \vec{n} \right\} \setminus \left\{ 2^{l+1}\vec{n} \right\},\ 0 \le l < L \tag{21}$$

Here, each set I_l contains the coordinates of the samples of the corresponding scale level $\mathbf{X}_l$.

We compress the scale levels $\mathbf{X}_l$ one by one, in order $\mathbf{X}_{L-1}, \mathbf{X}_{L-2}, ..., \mathbf{X}_1, \mathbf{X}_0$. We interpolate samples of each scale level $\mathbf{X}_l$, based on samples of less down-sampled scale levels $\mathbf{X}_{l+m}$. Then we quantize and encode the interpolation errors.

We use samples of all scale levels $\mathbf{X}_{l+m}$ to interpolate samples of each scale level $\mathbf{X}_l$, since scale levels $\mathbf{X}_{l+m}$ together compose the regular D-dimensional grid of signal samples with the step 2^{l+1}:

$$\left\{ x^{(l+1)}(\vec{n}) \right\} = \left\{ x\left(2^{l+1}\vec{n}\right) \right\} = \bigcup_{m=1}^{L-1} \mathbf{X}_{l+1} \tag{22}$$

Therefore, the hierarchical compression of each scale level is reduced to the interpolation compression of the signal by the signal described in the previous section. Thus, we optimize the decision rule twice for each scale level with hierarchical compression.

Thus, we reduce the hierarchical compression of each scale level $\mathbf{X}_l$ to the interpolation compression of the signal $x^{(l)}(\vec{n})$ by the signal $x^{(l+1)}(\vec{n}) = x^{(l)}(2\vec{n})$ described in the previous section. Therefore, we optimize the decision rule twice for each scale level during hierarchical compression.

IV. Experimental study of the adaptive interpolator during compression

We performed experimental studies of the adaptive interpolator as part of the hierarchical method and interpolation method of multidimensional signals compression.

We used a uniform scale with step $(2\varepsilon + 1)$ to quantize (4) the post-interpolation residues $\lambda(\vec{n})$ in both of these compression methods. We describe the quantizer Q and the dequantizer Q^{-1} when using this scale as follows:

$$Q\left(\lambda(\vec{n})\right) = \mathrm{int}\left(\frac{\varepsilon + \lambda(\vec{n})}{2\varepsilon + 1} \right) sign\left(\lambda(\vec{n})\right)$$
$$Q^{-1}\left(\lambda(\vec{n})\right) = q(\vec{n})(2\varepsilon + 1) \tag{23}$$

Here ε sets the controlled maximum error $\varepsilon = \max_{\vec{n}} \left| x(\vec{n}) - y(\vec{n}) \right|$ between the original $x(\vec{n})$ and decompressed $y(\vec{n})$ signals.

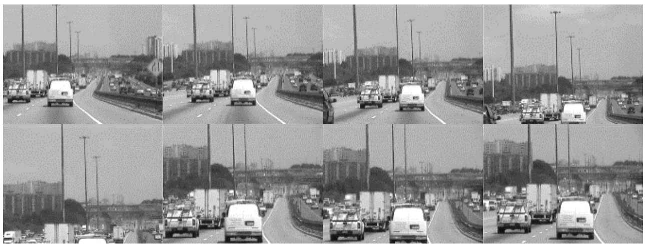

Fig. 1. Some frames of the test video signal "highway".

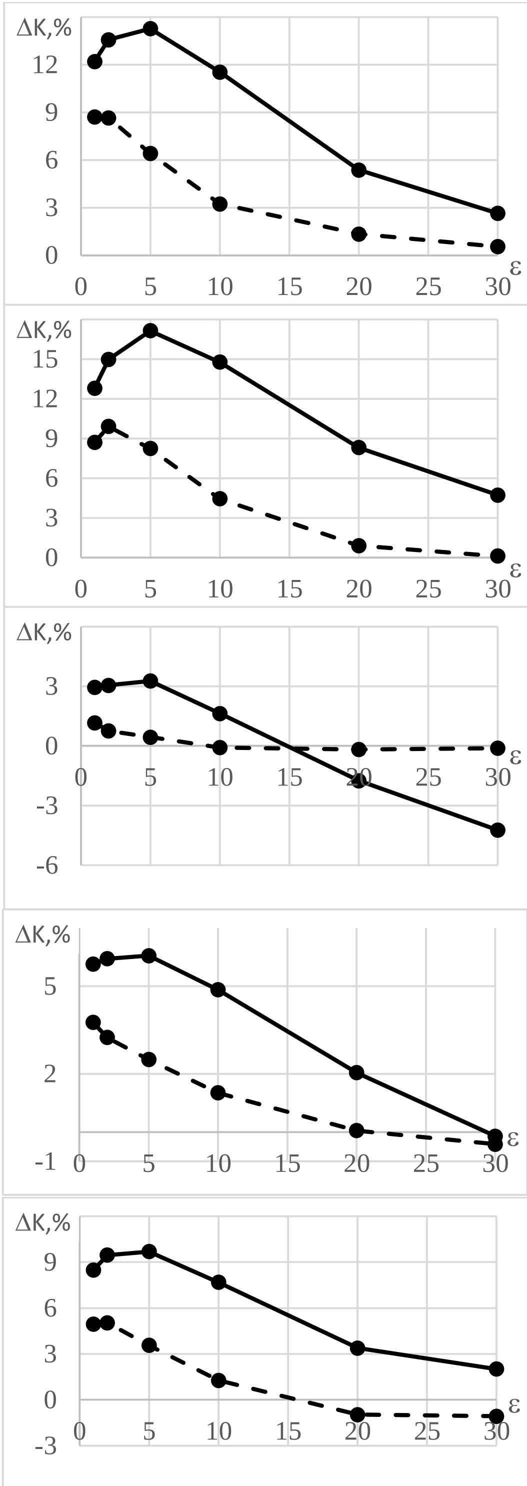

Fig. 2. The gain of the adaptive interpolator from the averaging interpolator within the hierarchical compression method (solid line) and the interpolation compression method (dashed line) in the signals: “escalator”, “beach”, “forest fire”, “building collapse”, “highway”.

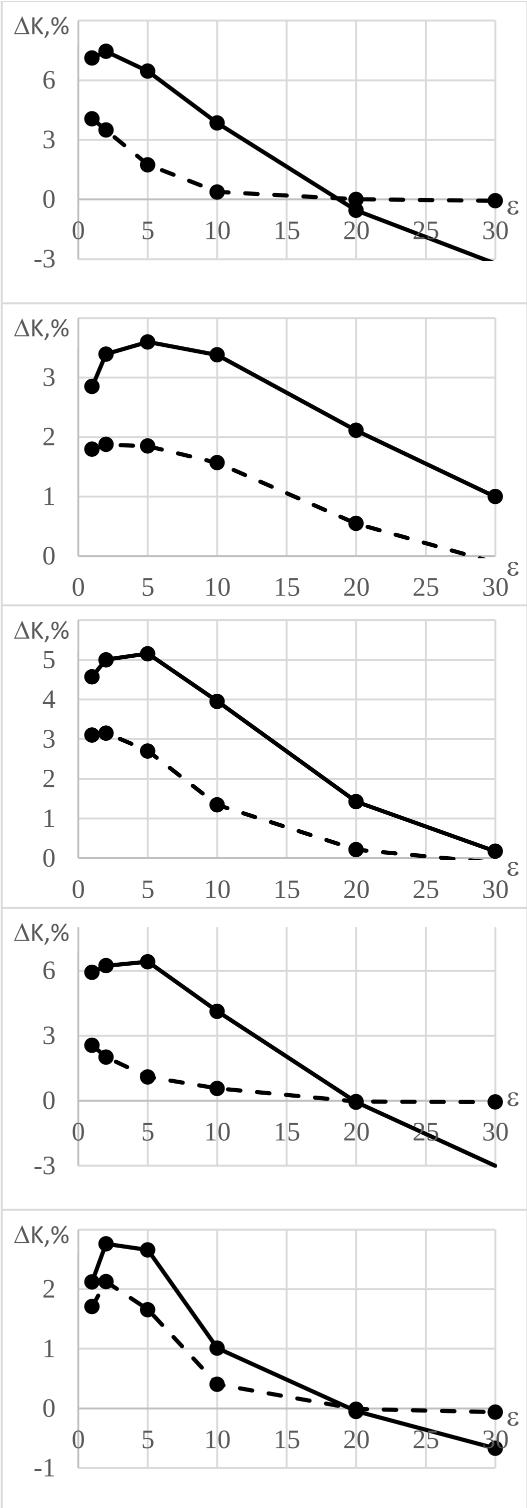

Fig. 3. The gain of the adaptive interpolator from the averaging interpolator within the hierarchical compression method (solid line) and the interpolation compression method (dashed line) in the signals: “fountain”, “marathon”, “railway”, “clouds”, “falling trees”.

We used natural test video signals from the dataset “Dynamic scenes data set” [15] (see the example in Fig. 1). We calculated the relative gain $\Delta K = \left(1 - K / \tilde{K}\right) \cdot 100\%$ in the compression coefficient, achieved by replacing the averaging interpolator (6) with the adaptive interpolator (here $\tilde{K}$, K are the compression coefficients with averaging or adaptive interpolator, respectively).

We show the dependence of the gain ΔK on the maximum error ε for several test signals in Fig. 2-3. The graphs confirm that the adaptive interpolator can significantly (up to 17%) increase the efficiency of hierarchical methods and interpolation methods of multidimensional signals compression.

V. Conclusion

We have adapted parameterized algorithms for interpolating multidimensional signals for hierarchical compression methods and interpolation compression methods based on the coding of quantized post-interpolation residues.

We proposed interpolation functions based on interpolation along the most preferred directions. We have chosen the criterion for decision rule optimization based on minimizing the entropy of the quantized post-interpolation residues. We have solved the problem of optimizing the decision rule by this criterion.

We performed computational experiments in natural multidimensional signals. These experiments confirmed the significant increase in the effectiveness of the considered compression methods using parameterized interpolators.

Acknowledgment

The reported study was funded by RFBR according to the research projects 18-01-00667 (in parts II.A, II.C, II.D, II.E, II.F, III, IV, V), 18-07-01312 (in part II.B), and the RF Ministry of Science and Higher Education within the state project of FSRC “Crystallography and Photonics” RAS under agreement 007-GZ/Ch3363/26 (in part I).

References

[1] K. Sayood, “Introduction to data compression,” The Morgan Kaufmann series in multimedia information and systems, 2012, 743 p.

[2] V. Gupta, V. Sharma and A. Kumar, “Enhanced image compression using wavelets,” International journal of research in engineering and science (IJRES), vol. 2, no. 5, pp. 55-62, 2014.

[3] J. Li, “Image compression: the mathematics of JPEG-2000,” Modern signal processing, vol. 46, pp. 185-221, 2003.

[4] W. Woon, A. Ho, T. Yu, S. Tam, S. Tan and L. Yap, “Achieving high data compression of self-similar satellite images using fractal,” IEEE international geoscience and remote sensing symposium (IGARSS), pp. 609-611, 2000.

[5] G. Wallace, “The JPEG still picture compression standard,” Communications of the ACM, vol. 34, no. 4, pp. 30-44, 1991.

[6] G. Plonka and M. Tasche, “Fast and numerically stable algorithms for discrete cosine transforms,” Linear algebra and its applications, vol. 394, no. 1, pp. 309-345, 2005.

[7] D. Vatolin, A. Ratushnyak, M. Smirnov and V. Yukin, “Data compression methods. The device of archivers, image and video compression,” M: Dialog-Mephi, 2002, 384 p.

[8] V. Soifer, “Methods of computer image processing,” M: Fizmatlit, 2003, 784 p.

[9] V. Sergeev, N. Glumov and M. Gashnikov, “Information technology for image compression in online remote sensing systems,” Bulletin of the Samara scientific center of the Russian Academy of Sciences, vol. 1, pp. 99-107, 1999.

[10] M. Gashnikov, “Interpolation based on context modeling for hierarchical compression of multidimensional signals,” Computer Optics, vol. 42, no. 3, pp. 468-475, 2018. DOI: 10.18287/2412-6179-2018-42-3-468-475.

[11] J. Woods, “Multidimensional signal, image, and video processing and coding,” Academic press, 2011, 616 p.

[12] R. Gonzalez and R. Woods, “Digital image processing,” NJ: Prentice Hall, 2007, 976 p.

[13] A. Maksimov and M. Gashnikov, “Adaptive interpolation of multidimensional signals for differential compression,” Computer

Optics, vol. 42, no. 4, pp. 679-687, 2018. DOI: 10.18287/2412-6179-2018-42-4-679-68.

[14] A. Maksimov and M. Gashnikov, “Parameter space dimension reduction for multidimensional signals differential pulse-code modulation interpolator,” 7th International symposium on digital forensics and security (ISDFS), vol. 8757494, 2019.

[15] C. Feichtenhofer, A. Pinz1 and R. Wildes, “Dynamic scenes data set,” URL: http://vision.eecs.yorku.ca/research/dynamic-scenes.