

# Interpolation of multidimensional signals using the reduction of the dimension of parametric spaces of decision rules

M V Gashnikov<sup>1,2</sup>

<sup>1</sup>Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086

<sup>2</sup>Image Processing Systems Institute of RAS - Branch of the FSRC "Crystallography and Photonics" RAS, Molodogvardejskaya street 151, Samara, Russia, 443001

e-mail: mgash@smr.ru

**Abstract.** In this paper, we consider the interpolation of multidimensional signals problem. We develop adaptive interpolators that select the most appropriate interpolating function at each signal point. Parameterized decision rule selects the interpolating function based on local features at each signal point. We optimize the adaptive interpolator in the parameter space of this decision rule. For solving this optimization problem, we reduce the dimension of the parametric space of the decision rule. Dimension reduction is based on the parameterization of the ratio between local differences at each signal point. Then we optimize the adaptive interpolator in parametric space of reduced dimension. Computational experiments to investigate the effectiveness of an adaptive interpolator are conducted using real-world multidimensional signals. The proposed adaptive interpolator used as a part of the hierarchical compression method showed a gain of up to 51% in the size of the archive file compared to the smoothing interpolator.

## 1. Introduction

Currently, the need to use multidimensional digital signals is becoming more acute [1]. It is primarily about such areas as remote sensing [2-3], processing of multispectral and hyperspectral signals [4], as well as video processing.

Now we know a large number of interpolation and approximation algorithms for such signals [1-16], and the high-performance requirements often do not allow us to use trivial linear, bilinear or bicubic interpolators [1]. In other words, there is a tendency to the more and more widespread use of more complex interpolation methods, such as the support vector method [5], locally optimal well-adapted basis functions [6], approximation by multidimensional orthogonal polynomials [7], multidimensional approximation and interpolation [8] etc.

However, the presence of a large number of well-developed solutions did not stop research in the development and modification of interpolation and approximation algorithms for multidimensional signals. The approximation based on Kronecker bases [9], splines [10], and tensors [11] continues to be improved. Artificial neural networks [7,12] are also increasingly used for interpolating signals. Even the well-known least squares method (OLS) continues to be modified [13] in recent years. In the foreign literature, special attention is paid to the sparse approximation method [14], which is the basis for the "compressed sensing" approach [15-16].

All the above algorithms have a sufficiently high accuracy in solving the corresponding applied problems. However, these algorithms have high computational complexity. In this paper, we propose fast interpolation algorithms for multidimensional signals that are adaptive due to automatic switching between interpolating functions at each point of the signal. This adaptability makes it possible to ensure high interpolation accuracy with low computational complexity, due to the simplicity of the interpolating functions used.

The proposed interpolators are parameterized. Therefore, we can optimize them according to various criteria, the choice of which is determined by the specifics of the applied problem being solved. Optimization of adaptive interpolators is carried out in the space of their parameters. The complexity of this optimization is substantial if resources are limited. In this paper, we propose an algorithm for reducing the complexity of optimization due to the dimension reduction of the interpolator parametric space.

## 2. Adaptive interpolation of multidimensional signals

Let  $C(\vec{x})$  be a multidimensional digital signal, and  $\vec{x}$  be the vector of arguments. Let an arbitrary count  $C(\vec{x})$  be necessary to interpolate using the nearest reference samples  $\{\hat{C}_k(\vec{x})\}$ . Let  $\{\mathbf{P}^{(i)}(\{\hat{C}_k(\vec{x})\})\}$  be the set of interpolation functions used. Thus, for the current sample  $C(\vec{x})$  several interpolating values can be calculated:

$$P_i(\vec{x}) = \mathbf{P}^{(i)}(\{\hat{C}_k(\vec{x})\}) \quad (1)$$

The parameterized decision rule  $R$  performs the choice of the interpolating value for each signal sample:

$$P(\vec{x}) = P^{(i)}(\vec{x}), \quad i = R(\eta(\vec{x}), \eta^{\text{lim}}), \quad (2)$$

Rule  $R$  uses a local feature vector  $\eta(\vec{x})$ , which is calculated based on the nearest reference samples  $\{C_k(\vec{x})\}$ . Let the decision rule be parameterized, i.e. depends on the parameter  $\eta^{\text{lim}}$ . The value of this parameter is determined by optimizing a specific criterion, which depends on the applied task. This criterion can be, for example, the criterion for minimizing the energy of post-interpolation residues:

$$\delta(\eta^{\text{lim}}) = \sum_{\vec{x}} |f(\vec{x})| \rightarrow \min_{\eta^{\text{lim}}}, \quad f(\vec{x}) = C(\vec{x}) - P(\vec{x}), \quad (3)$$

where  $f(\vec{x})$  are post-interpolation residues.

The criterion for minimizing the energy of post-interpolation residues can be used, in particular, when solving the problem of matching [17-18] of heterogeneous signals that differ in resolution, number of components, etc.

In this paper, we propose to use the criterion for minimizing the entropy [19] of post-interpolation residues. This criterion is more suitable for the problem of signal compression than the criterion considered above. When using the criterion for minimizing entropy, one should take into account that post-interpolation residues are quantized before statistical coding in many compression methods, for example, in differential [20-21] and hierarchical [22-23] compression methods. In this paper, the quantizer with a uniform [19] scale is used to calculate quantized post-interpolation residues  $q(\vec{x})$ :

$$q(\vec{x}) = \left[ \left( f(\vec{x}) + \varepsilon_{\text{max}} \right) / (2\varepsilon_{\text{max}} + 1) \right] \text{sign}(f(\vec{x})), \quad (4)$$

where  $[\dots]$  means the selection of the integer part of the number,  $\varepsilon_{\text{max}}$  is the maximum error [20] when quantizing.

Thus, when using the specified "entropy" optimization criterion for the interpolator, it is necessary to minimize the entropy  $H$  of the quantized post-interpolation residues  $q(\vec{x})$ :

$$H(\eta^{\text{lim}}) = - \sum_{q=-M+1}^{M-1} N_q(\eta^{\text{lim}}) \ln N_q(\eta^{\text{lim}}) \rightarrow \min_{\eta^{\text{lim}}}, \quad M = \max_{\bar{x}}(C(\bar{x})), \quad (5)$$

where  $N_q(\eta^{\text{lim}})$  is the number of quantized post-interpolation residues equal to  $q$ , and  $M$  is the maximum value of the original signal.

### 3. Reduction of the parametric space dimension

With significant constraints on computational complexity, interpolation algorithms are often used [1,4], which use “smoothing” (averaging) over some a set of nearest reference samples:

$$P(\bar{x}) = \frac{1}{N} \sum_{k=1}^N \hat{C}_k(\bar{x}), \quad (6)$$

where  $\hat{C}_k(\bar{x})$  are the nearest reference samples,  $N$  is the number of these samples.

The specific of the interpolator application determines the arrangement of these reference samples. This arrangement can be quite non-trivial (see below) for some tasks related, for example, to the use of some image compression methods.

As mentioned above, the use of such simple interpolation algorithms is typical in situations where it is necessary to minimize the computational complexity. In particular, interpolators of this type are used in differential [20-21] and hierarchical [22-23] compression methods for multidimensional signals.

The “smoothing” interpolation algorithm is sufficiently accurate on smoothly varying signal regions since averaging reduces noise. However, the “smoothing” interpolator is always characterized by an increase in the interpolation error at the boundaries of the indicated smoothly varying regions (i.e., at the boundaries). To interpolate such boundaries, we can use algorithms that use the so-called interpolation “along the border”. For a two-dimensional signal, in particular, Graham's nonlinear interpolation algorithm [20] works in this way.

When using this algorithm, the Interpolated value is equal to the reference signal sample to which the local boundary is directed. However, this algorithm, for obvious reasons, has less accuracy on smoothly varying parts of the signal.

In this article, we propose an adaptive parameterized interpolation algorithm that combines the advantages of both the described approaches: “smoothing” approach and “boundary” approach. The proposed interpolation algorithm is based on the approach described in Section 2. The proposed algorithm automatically switches between “smoothing” and “boundary” interpolators, depending on how sharp the boundary is in the local neighborhood of the processed sample.

Next, we describe the proposed adaptive interpolator. We also specify the interpolating functions (1) and decision rule (2). Denote by  $N_c$  the number of boundary directions taken into account. Let  $\{\lambda_i(\bar{x}): 0 \leq i < N_c\}$  be the set of averaged absolute values of differences between the reference samples  $\hat{C}_k(\bar{x})$  in each of the directions under consideration:

$$\lambda_i(\bar{x}) = \left| \hat{C}_t(\bar{x}) - \hat{C}_\tau(\bar{x}) \right| \quad (7)$$

where  $t$  and  $\tau$  are the indices of the reference samples.

The differences  $\lambda_i(\bar{x})$  determine the presence and intensity of the boundary in the local neighborhood of the current signal sample. We can use several thresholds  $\lambda_i^{\text{lim}}$  to decide on the presence and direction of the boundary. We compare the described differences  $\lambda_i$  with these threshold values  $\lambda_i^{\text{lim}}$ . If there is no boundary in the neighborhood of the current signal sample, then we use a “smoothing” interpolating function of the form (6):

$$P(\vec{x}) = P^{(1)}(\vec{x}) = \frac{1}{N} \sum_{k=1}^N \hat{C}_k(\vec{x}), \text{ if } \lambda_i \leq \lambda_i^{\text{lim}}, i \in [0, N_c) \quad (8)$$

If there is a boundary in the local neighborhood, then for interpolation we use the average value  $\tilde{C}_j(\vec{x})$  of the two nearest reference samples located in the direction of the local boundary:

$$P(\vec{x}) = P^{(2)}(\vec{x}) = \tilde{C}_j(\vec{x}), \text{ if } \lambda_i > \lambda_i^{\text{lim}}, i \in [0, N_c) \quad (9)$$

Thus, we need to solve an optimization problem in a  $N_c$ -dimensional parametric space to find the best thresholds  $\lambda_i^{\text{lim}}$ .

The application determines the dimension  $N_c$  of the parameter space  $\lambda_i^{\text{lim}}$ . As we show below, in the problem of hierarchical compression [22-23]  $N_c = 2$  for a two-dimensional signal,  $N_c = 4$  for a three-dimensional signal (in the simplest case), and then  $N_c$  grows rapidly with increasing signal dimension. However, the search for parameters can be an overly time-consuming task during compression even when  $N_c = 2$ .

In this paper, we propose to reduce the dimension of the parameter space  $\lambda_i^{\text{lim}}$  to reduce the computational complexity of finding these parameters. For this, we propose to use not the absolute values of the differences  $\lambda_i$ , but their ratio during interpolation. If there is no boundary (or the boundary is weak) in the neighborhood of the current sample, all differences have close values. If there is a clear boundary in this neighborhood, then the smallest difference  $\lambda_j$  corresponds to the direction of this boundary:

$$j(\vec{x}) = \arg \min_i \{ \lambda_i(\vec{x}) \}$$

Moreover, if there is a boundary in the neighborhood, this difference is significantly different from all other differences, including the nearest difference  $\lambda_r$ :

$$r(\vec{x}) = \arg \min_{i: i \neq j} \{ \lambda_i(\vec{x}) \}$$

Based on this reasoning, in this article the feature of the boundary direction is defined as the difference between the two smallest differences  $\lambda_i$ :

$$\eta(\vec{x}) = \lambda_r(\vec{x}) - \lambda_j(\vec{x}) \quad (10)$$

When interpolating each sample, the feature  $\eta(\vec{x})$  is compared with a threshold  $\eta^{\text{lim}}$ . If the feature  $\eta(\vec{x})$  is small enough (less than the threshold  $\eta^{\text{lim}}$ ), then it is considered that there is no boundary in the neighborhood of the current sample. Therefore, a ‘‘smoothing’’ interpolation of the form (8) is used:

$$P(\vec{x}) = P^{(1)}(\vec{x}) = \frac{1}{N} \sum_{k=1}^N \hat{C}(\vec{x}_k), \text{ if } \eta(\vec{x}) \leq \eta^{\text{lim}} \quad (11)$$

The difference from (8) is that now this interpolation is one-parameter. If the feature  $\eta(\vec{x})$  is larger than the threshold  $\eta^{\text{lim}}$ , then interpolation of the form (9) ‘‘along the boundary’’ is performed:

$$P(\vec{x}) = P^{(2)}(\vec{x}) = \hat{C}_j(\vec{x}), j = \arg \min_i \{ \lambda_i \}, \text{ if } \eta(\vec{x}) > \eta^{\text{lim}}. \quad (12)$$

Thus, by reducing the dimension of the parameter space, we reduce the multiparameter optimization problem to a one-parameter one, in which the only parameter is  $\eta^{\text{lim}}$ . An automatic optimization procedure similar to [23] is used to calculate this parameter.

#### 4. Multidimensional adaptive interpolator in the problem of hierarchical compression

The proposed multidimensional adaptive interpolator can be used in various signal processing tasks. In particular, in this paper, we consider the use of this interpolator in the problem of compression. As an example of the compression method, we consider the hierarchical compression method [22-23].

This method uses a unique hierarchical non-redundant representation (see Figures 1-2) of the original multidimensional signal  $\mathbf{C} = \{C(\vec{x})\}$  as a set of  $L$  scale levels  $\mathbf{C}_l$ :

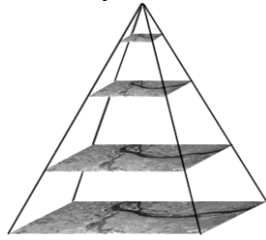
$$\mathbf{C} = \bigcup_{l=0}^{L-1} \mathbf{C}_l, \quad \mathbf{C}_l = \{C_l(\vec{x})\} = \{C(\vec{x}) : \vec{x} \in I_l\}, \quad (17)$$

where  $I_l$  is the set of sample indices of the corresponding scale level  $\mathbf{X}_l$ :

$$I_{L-1} = \{2^{L-1}\vec{x}\}, \quad I_l = \{2^l\vec{x}\} \setminus \{2^{l+1}\vec{x}\}, \quad 0 \leq l < L \quad (18)$$

Thus, the most resampled scale level  $\mathbf{C}_{L-1}$  is a "grid" of signal samples with step of  $2^{(L-1)}$ , and all other scale levels with the numbers  $l = (L-1), (L-2), \dots, 1, 0$  are grids of signal samples with the step of  $2^l$ , of which samples are removed with the step of  $2^{l+1}$ .

With hierarchical compression, the scale levels of the signal are compressed sequentially, from the most resampled level  $\mathbf{C}_{L-1}$  to the least resampled level  $\mathbf{C}_0$ . In this case, samples of more resampled levels are used for interpolation of samples of less resampled levels.



**Figure 1.** Hierarchical representation of a two-dimensional signal by four scale levels.

3	1	2	1	3
1	1	1	1	1
2	1	2	1	2
1	1	1	1	1
3	1	2	1	3

**Figure 2.** Level numbers in the hierarchical representation of the signal (the level number is zero in the empty cells).

Most often, to reduce computational complexity with hierarchical compression, a smoothing interpolator of the form (6) is used. In the three-dimensional case, this interpolator can be written as:

$$P_i^{(1)}(2m+1, 2n+1, 2k+1) = \frac{1}{8} \sum_{m'=0}^1 \sum_{n'=0}^1 \sum_{k'=0}^1 C_{l+1}(m+m', n+n', k+k') \quad (19)$$

Differences  $\lambda_i$  (7) in this case take the form:

$$\lambda_i^{(0)}(2m+1, 2n+1, 2k+1) = |C_{l+1}(m+1, n+1, k+1) - C_{l+1}(m, n, k)|, \quad (20)$$

$$\lambda_i^{(1)}(2m+1, 2n+1, 2k+1) = |C_{l+1}(m, n+1, k+1) - C_{l+1}(m+1, n, k)|, \quad (21)$$

$$\lambda_i^{(2)}(2m+1, 2n+1, 2k+1) = |C_{l+1}(m, n, k+1) - C_{l+1}(m+1, n+1, k)|, \quad (22)$$

$$\lambda_i^{(3)}(2m+1, 2n+1, 2k+1) = |C_{l+1}(m+1, n, k+1) - C_{l+1}(m, n+1, k)|. \quad (23)$$

Thus, the adaptive three-dimensional interpolator allows us to automatically switch between smoothing interpolation (19) and interpolation along the boundary of one of the four directions shown in Fig. 3. In other words, at each point of the signal we can use the interpolating value (19) or one of the following four interpolating values:

$$V_i^{(0)}(2m+1, 2n+1, 2k+1) = \frac{1}{2} (C_{l+1}(m+1, n+1, k+1) + C_{l+1}(m, n, k)), \quad (24)$$

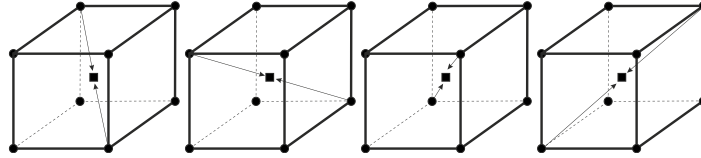
$$V_l^{(1)}(2m+1, 2n+1, 2k+1) = \frac{1}{2}(C_{l+1}(m, n+1, k+1) + C_{l+1}(m+1, n, k)) \quad (25)$$

$$V_l^{(2)}(2m+1, 2n+1, 2k+1) = \frac{1}{2}(C_{l+1}(m, n, k+1) + C_{l+1}(m+1, n+1, k)) \quad (26)$$

$$V_l^{(3)}(2m+1, 2n+1, 2k+1) = \frac{1}{2}(C_{l+1}(m+1, n, k+1) + C_{l+1}(m, n+1, k)) \quad (27)$$

The “boundary” interpolating function (12) in this situation takes the form:

$$P_l^{(2)}(2m+1, 2n+1, 2k+1) = V_l^{(j)}(2m+1, 2n+1, 2k+1), j = \arg \min_i \{\lambda_i(2m+1, 2n+1, 2k+1)\} \quad (28)$$



**Figure 3.** Three-dimensional interpolating functions (24-27) of the adaptive interpolator.

Thus, in the introduced notation, the three-dimensional adaptive interpolator is described by the expression:

$$P_l(2m+1, 2n+1, 2k+1) = \begin{cases} P_l^{(1)}(2m+1, 2n+1, 2k+1), & \text{if } \eta_l(2m+1, 2n+1, 2k+1) \leq \eta_l^{\lim} \\ P_l^{(2)}(2m+1, 2n+1, 2k+1), & \text{if } \eta_l(2m+1, 2n+1, 2k+1) > \eta_l^{\lim} \end{cases} \quad (29)$$

where, for each scale level, the feature  $\eta_l(2m+1, 2n+1, 2k+1)$  is calculated according to expressions (8-10), and compared with threshold  $\eta_l^{\lim}$  for each level.

In the two-dimensional case, the smoothing interpolator (6) can be written in the form:

$$P_l^{(1)}(2m+1, 2n+1) = \frac{1}{4}(C_{l+1}(m, n) + C_{l+1}(m+1, n) + C_{l+1}(m, n+1) + C_{l+1}(m+1, n+1)) \quad (30)$$

Differences  $\lambda_i$  (7) in this case take the form:

$$\lambda_l^{(0)}(2m+1, 2n+1) = |C_{l+1}(m, n) - C_{l+1}(m+1, n+1)| \quad (31)$$

$$\lambda_l^{(1)}(2m+1, 2n+1) = |C_{l+1}(m, n+1) - C_{l+1}(m+1, n)| \quad (32)$$

The corresponding interpolating values are written as follows:

$$V_l^{(0)}(2m+1, 2n+1) = \frac{1}{2}(C_{l+1}(m+1, n+1) + C_{l+1}(m, n)) \quad (33)$$

$$V_l^{(1)}(2m+1, 2n+1) = \frac{1}{2}(C_{l+1}(m, n+1) + C_{l+1}(m+1, n)) \quad (34)$$

For the interpolation itself in the two-dimensional case, it is more appropriate to use a two-parameter adaptive interpolation function:

$$P_l(2m+1, 2n+1) = \begin{cases} V_l^{(0)}(2m+1, 2n+1), & \text{if } \eta_l(2m+1, 2n+1) < \eta_l^{\lim(-)} \\ P_l^{(1)}(2m+1, 2n+1), & \text{if } \eta_l^{\lim(-)} \leq \eta_l(2m+1, 2n+1) \leq \eta_l^{\lim(+)} \\ V_l^{(1)}(2m+1, 2n+1), & \text{if } \eta_l(2m+1, 2n+1) > \eta_l^{\lim(+)} \end{cases} \quad (35)$$

where  $\eta_l(m, n)$  is the feature of boundary direction.

$$\eta_l(m, n) = \lambda_l^{(0)}(m, n) - \lambda_l^{(1)}(m, n) \quad (36)$$

At each signal point, the feature  $\eta_l(m, n)$  is compared with two thresholds  $\eta_l^{\lim(-)}$ ,  $\eta_l^{\lim(+)}$ , because optimization by these parameters can be done separately.

### 5. Experimental study of the adaptive interpolator for real-world multidimensional signals

The proposed adaptive interpolator (30-36) was implemented programmatically in C++ and built into the hierarchical compression method. This software implementation was used in this article to study the effectiveness of the proposed interpolator. To this end, computational experiments were performed in real-world multidimensional signals (see Figure 4-5) of two signal sets:

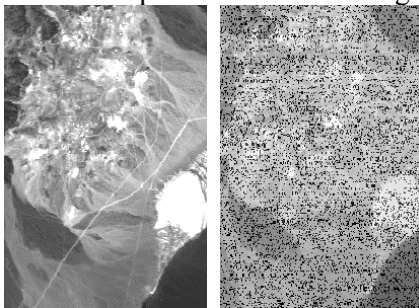
set no. 1 - TokyoTech hyperspectral dataset [24] (signal sizes 500x500x31 samples, 13 bits)

set no. 2 - AVIRIS hyper-spectrometer [25] (signal sizes 1086x614x224 samples, 16 bits).

A measure of the effectiveness of the proposed interpolator was the relative gain in the archive size, which was achieved by replacing the smoothing interpolator with an adaptive interpolator in the frame of hierarchical compression method:

$$\Delta = (K_{adapt} / K_{smooth} - 1) \cdot 100\% \quad (37)$$

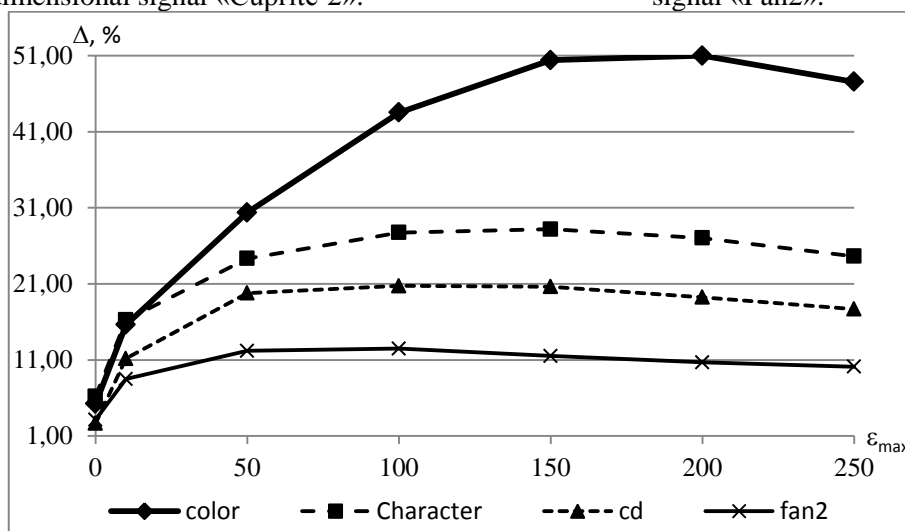
where  $K_{smooth}$ ,  $K_{adapt}$  are the compression ratios of the hierarchical compression method when using a smoothing and adaptive interpolator, respectively. Typical results of computational experiments are shown in Fig. 6-7 and Table 1. The adaptive interpolator has a gain of up to 51% in the size of the archive file compared to the smoothing interpolator.



**Figure 4.** Fragments of bands 10, 86 of test multidimensional signal «Cuprite-2».



**Figure 5.** Bands 0, 30 of test multidimensional signal «Fan2».



**Figure 6.** Gain Δ (in %) the adaptive interpolator for the averaging interpolator in test signals «Color», «Character», «CD», «Fan2».

### 6. Conclusion

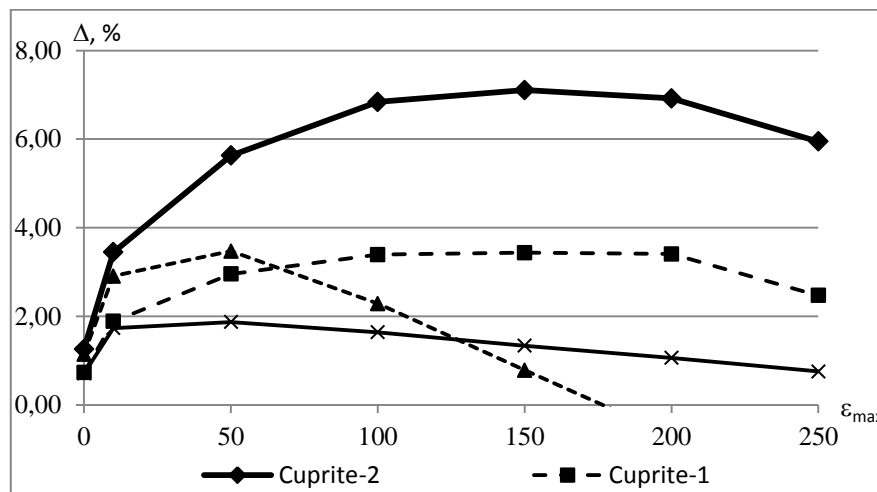
We proposed interpolation algorithms for multidimensional signals based on automatic switching between simple interpolating functions at each point of the signal. We described the parameterized decision rules that perform this switch. We optimized the parameters of these decision rules on the criteria for the minimum energy of post-interpolation residues and criteria of the minimum entropy of quantized post-interpolation residues. We proposed a method for reducing the dimension of the parametric space of decision rules. We performed computational experiments to study the proposed

interpolators on real-world multidimensional signals. We experimentally proved that using the adaptive interpolator instead of a smoothing one can significantly improve the efficiency of hierarchical signal compression.

**Table 1.** The gain  $\Delta$  (in %) of the adaptive interpolator for the averaging interpolator depending on maximum error  $\varepsilon_{\max}$  in test signals «Butterfly2», « Butterfly3», ... ,«Low Altitude», «Moffett Field».

$\varepsilon_{\max}$	Signal set	0	10	50	100	150	200	250	max( $\Delta$ )
Butterfly2	set no.1	1.17	4.84	8.00	9.17	9.57	9.19	8.62	9.57
Butterfly3	set no.1	1.87	7.24	13.81	17.07	17.74	16.74	14.98	17.74
Butterfly4	set no.1	1.87	6.37	11.45	13.17	13.81	13.51	12.98	13.81
Butterfly5	set no.1	1.30	6.36	11.35	12.63	12.47	11.00	9.41	12.63
Butterfly6	set no.1	1.23	5.20	9.01	10.90	11.58	11.28	11.10	11.58
Butterfly7	set no.1	1.48	5.24	9.81	11.55	12.16	11.55	10.39	12.16
Butterfly	set no.1	1.68	5.49	9.02	9.89	9.49	8.64	7.64	9.89
Butterfly8	set no.1	1.88	7.03	11.46	12.37	12.36	12.14	10.72	12.37
cd	set no.1	2.66	11.18	19.79	20.74	20.64	19.25	17.69	20.74
Character	set no.1	6.21	16.26	24.34	27.73	28.19	27.02	24.64	28.19
Chart24	set no.1	5.57	18.18	36.27	40.18	36.60	30.42	25.57	40.18
ChartRes	set no.1	7.80	18.76	28.57	36.31	39.30	39.49	38.91	39.49
Cloth2	set no.1	0.64	1.62	2.28	2.13	1.92	1.48	1.14	2.28
Cloth3	set no.1	1.16	2.88	4.11	4.28	4.18	2.98	1.88	4.28
Cloth4	set no.1	0.58	1.32	2.11	2.18	1.51	0.66	0.06	2.18
Cloth5	set no.1	2.10	4.60	6.42	5.91	5.66	5.11	4.07	6.42
Cloth	set no.1	0.84	1.94	2.92	2.74	2.40	1.81	1.27	2.92
colorchart	set no.1	4.10	15.40	37.31	44.07	39.92	34.67	30.61	44.07
color	set no.1	5.24	15.63	30.38	43.53	50.40	51.00	47.56	51.00
doll	set no.1	0.44	1.34	1.72	1.82	1.83	1.87	1.78	1.87
fan2	set no.1	3.15	8.50	12.19	12.49	11.51	10.68	10.11	12.49
fan3	set no.1	2.50	6.54	9.77	9.92	9.41	8.82	8.23	9.92
fan	set no.1	1.76	4.12	6.29	6.25	5.51	4.70	3.99	6.29
flower2	set no.1	1.60	4.35	5.63	4.59	2.75	1.24	-0.71	5.63
flower3	set no.1	2.28	6.74	9.43	8.57	6.85	4.97	3.07	9.43
flower	set no.1	2.79	8.42	11.50	9.18	6.54	4.00	2.34	11.50
party	set no.1	3.18	9.51	15.22	16.70	17.03	16.46	16.45	17.03
tape2	set no.1	4.60	9.87	14.46	14.54	13.07	11.37	9.95	14.54
tape	set no.1	1.61	3.99	5.63	5.61	5.22	4.64	4.11	5.63
Tshirts2	set no.1	0.76	2.32	3.96	4.77	4.65	4.48	4.52	4.77
Tshirts	set no.1	0.75	2.26	3.82	4.59	4.88	5.11	5.28	5.28
Cuprite-1	set no.2	0.73	1.89	2.96	3.40	3.44	3.41	2.47	3.44
Cuprite-2	set no.2	1.26	3.45	5.63	6.84	7.11	6.92	5.95	7.11
Low Altitude	set no.2	1.15	2.91	3.47	2.29	0.79	-0.69	-2.16	3.47
Moffett Field	set no.2	0.71	1.74	1.87	1.64	1.34	1.06	0.76	1.64





**Figure 7.** Gain  $\Delta$  (in %) the adaptive interpolator for the averaging interpolator in test signals «Cuprite-1», «Cuprite-2», «Low Altitude», «Moffett Field».

## 7. References

- [1] Woods J 2011 *Multidimensional Signal, Image, and Video Processing and Coding* (Academic Press) p 211
- [2] Jensen J 2007 *Remote sensing of the environment: an Earth resource perspective* (Prentice Hall) p 619
- [3] Campbell J 2002 *Introduction to remote sensing* (Guilford Press) p 667
- [4] Borengasser M, Hungate W and Watkins R 2007 *Hyperspectral remote sensing: Principles and applications* (CRC Press) p 128
- [5] Vapnik V 1998 *Statistical Learning Theory* (John Wiley & Sons)
- [6] Vasin Y and Neymark Y 1978 Adaptive compression recurrent algorithms using well-adapted local restoring functions *Mathematical software for CAD: Interuniversity collection* **13**
- [7] Gulakov K 2013 The choice of neural network architecture for solving the problems of approximation and regression analysis of experimental data *Bulletin of Bryansk State Technical University* **2** 95-105
- [8] Bakhvalov Y 2007 *The method of multidimensional interpolation and approximation and its applications* (M: Sputnik+) p 108
- [9] Caiafa C 2016 Computing Sparse Representations of Multidimensional Signals Using Kronecker Bases *Neural Computation Volume* **25** 186-220
- [10] Butyrsky E, Kuvaldin I and Chalkin V 2010 Approximation of multidimensional functions *Scientific instrument-making* **20** 82-92
- [11] Cobanu M and Makarov D 2014 Compression of images by tensor approximation *Problems of development of promising micro- and nanoelectronic systems* 109-112
- [12] Gulakov K 2016 Modeling multidimensional objects on the basis of cognitive maps with neural network identification of parameters *Thesis for Ph.D*
- [13] Cohen A, Davenport M and Leviatan D 2013 On the stability and accuracy of least squares approximations *Comput. Math.* **13** 819-834
- [14] Sahnoun S, Djermoun E, Brie D and Comon P 2017 A simultaneous sparse approximation method for multidimensional harmonic retrieval *Signal Processing* **131** 36-48
- [15] Donoho D 2006 Compressed sensing *IEEE Trans. Inform. Theory* **52** 1289-1306
- [16] Bigot J, Boyer C and Weiss P 2016 An analysis of block sampling strategies in compressed sensing *IEEE Trans. Inform. Theory* **62** 2125-2139
- [17] German E 2014 Algorithms for combining heterogeneous images in on-board visualization systems *PhD Thesis*
- [18] Muratov E and Nikiforov M 2014 Methods to reduce the computational complexity of algorithms for combining heterogeneous images *Cloud of Science* **1**

- [19] Sayood K 2012 *Introduction to Data Compression* (The Morgan Kaufmann Series in Multimedia Information and Systems) p 743
- [20] Gonzalez R and Woods E 2007 *Digital Image Processing* (Prentice Hall) p 976
- [21] Maksimov A I and Gashnikov M V 2018 Adaptive interpolation of multidimensional signals for differential compression *Computer Optics* **42** 679-687 DOI: 10.18287/2412-6179-2018-42-4-679-687
- [22] Sergeev V, Gashnikov M and Glumov N 1999 The Informational Technique of Image Compression in Operative Remote Sensing Systems *RAS Samara Research Center Bulletin* **1** 99-107
- [23] Gashnikov M V 2018 Interpolation based on context modeling for hierarchical compression of multidimensional signals *Computer Optics* **42(3)** 468-475 DOI: 10.18287/2412-6179-2018-42-3-468-475
- [24] TokyoTech 31-band Hyperspectral Dataset URL: <http://www.ok.sc.e.titech.ac.jp/res/MSI/MSIdata31.html> (03.11.2018)
- [25] AVIRIS Data – Ordering Free AVIRIS Standard Data Products *Jet Propulsion Laboratory* URL: [http://aviris.jpl.nasa.gov/data/free\\_data.html](http://aviris.jpl.nasa.gov/data/free_data.html) (03.11.2018)

### **Acknowledgments**

The work was partly funded by RFBR according to the research project 18-01-00667 in parts of «2 Adaptive interpolation of multidimensional signals» – «5 Experimental study of the adaptive interpolator for real-world multidimensional signals» and by the Russian Federation Ministry of Science and Higher Education within a state contract with the "Crystallography and Photonics" Research Center of the RAS under agreement 007-Г3/Ч3363/26 in part of «1 Introduction».