

DEVELOPMENT OF AN ALGORITHM FOR AUTOMATIC CONSTRUCTION OF A COMPUTATIONAL PROCEDURE OF LOCAL IMAGE PROCESSING, BASED ON THE HIERARCHICAL REGRESSION

V.N. Kopenkov¹, V.V. Myasnikov^{1,2}

¹Samara National Research University, Samara, Russia,

²Image Processing Systems Institute of RAS, – Branch of the FSRC “Crystallography and Photonics” RAS, Samara, Russia

Abstract

In this paper, we propose an algorithm for the automatic construction (design) of a computational procedure for non-linear local processing of digital signals/images. The aim of this research is to work out an image processing algorithm with a predetermined computational complexity and achieve the best quality of processing on the existing data set, while avoiding a problem of retraining or doing with less training. To achieve this aim we use a local discrete wavelet transform for a preliminary image analysis and the hierarchical regression to construct a local image processing procedure on the basis of a training dataset. Moreover, we work out a method to decide whether the training process should be completed or continued. This method is based on the functional of full cross-validation control, which allows us to construct the processing procedure with a predetermined computational complexity and veracity, and with the best quality.

Keywords: local processing, hierarchical regression, computational efficiency, machine learning, precedent-based processing, functional of full cross-validation.

Citation: Kopenkov VN, Myasnikov VV. Development of an algorithm for automatic construction of a computational procedure of local image processing, based on the hierarchical regression. *Computer Optics* 2016; 40(5): 713-720. DOI: 10.18287/2412-6179-2016-40-5-713-720.

Acknowledgments: This work was financially supported by the Russian Science Foundation (RSF), grant no. 14-31-00014 “Establishment of a Laboratory of Advanced Technology for Earth Remote Sensing”.

Introduction

The tasks of image processing and signal analysis need to be solved in different fields of human activity [1 – 3]. Local processing of digital images is one of the most important kinds of transformation in the theory and practice of digital image processing and computer vision.

Historically, the first processing procedures used local linear methods that allow the construction of optimal (in some sense), processing procedures [1, 3]. However, the taking into consideration of new digital signal processing tasks (processing of video, audio, satellite images, etc.), problems of processing large amounts of information (satellite images, remote sensing data, hyperspectral data, multi-dimensional signals), processing in real time, and needs of rising of processing efficiency resulted in the necessity of using nonlinear type of transformations [1, 4]. One of the most common approach currently in use is the implementation of the cybernetic principle of the "black box" (the terms of other authors is the processing via recognition, processing basing on precedents and so on). In this case transformation itself and its parameters are determined by the analyzing of the input and output signals, or images.

The classic approach to construction of approximately universal procedures of local adaptive digital signal and images processing, which implements the principle of "black box" is based on usage of artificial neural networks technique [4]. An alternative, but substantially less researched version of described task solution based on using of a hierarchical computational structure, such as the decisions tree and regressions tree. [5, 6]. This paper develops the idea of the creation of universal mechanism for the construction of local non-linear computational processing procedures based on a hierarchical scheme and features based on local discrete wavelet decomposition of the image.

In addition, the methodology of decision-making on stopping the learning process, as well as the veracity of the obtained results are also presented in the article. Usually, for estimation of the generalization capability and selection a stop learning rule for processing procedure the Vapnik–Chervonenkis statistical theory is used [7]. This theory interconnects the three parameters of training: training error, veracity (reliability), and the length of training dataset. But estimates of statistical theory are highly overestimated and ignore the potential rearrangement of training and testing dataset elements. A more efficient way to estimate the generalization capability is the use of Vorontsov combinatorial theory [8], which is based on evaluation of the functional of full cross-validation, assuming verification all possible combinations of division dataset into training and testing parts. The correct solution to the task of construction of an image processing procedures which takes into account all combination of sets of training and control data is unrealizable in practice because of the giant search of variants of different combinations.

The paper is organized as follows. The first section devoted to subject introduction. The task specification and description of proposed solution, as well as a scheme of processing process are presented in the second section. Description and structure of the algorithm of construction of local image processing procedure and its parameters is presented in the third section. The fourth section is devoted to the description of methodology which allows to determine the rule for stop the process of formation and busting various combinations of training and control samples and stop construction process in general. And finally, conclusions, recommendations, acknowledgements, and references are presented in the end of the paper.

Image local processing model

Model of the local image processing technology, which implement the principle of "black box" (processing through the recognition or based on precedents), suggests decomposition of the transformation in two stages: the formation of the image fragment description (local features computing) and calculation of transformation results. The general scheme of image processing is shown in Fig 1.

To formalize the local image processing problem based on the proposed scheme let us introduce the following description of the pixel neighborhood:

$\Theta = \{(n_1, n_2), n_1 = \overline{0, N_1 - 1}, n_2 = \overline{0, N_2 - 1}\}$ is an image domain,

$x : \Theta \rightarrow \mathbf{R}$ is an input image,
 $(n_1, n_2) \rightarrow x(n_1, n_2)$

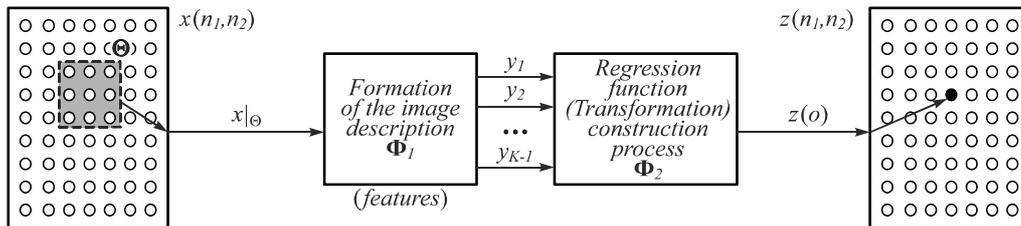


Fig. 1. A scheme of local image processing

These features are used to calculate the result of transformation $\Phi_2: \mathbf{R}^K \rightarrow \mathbf{K}$ (and to generate the resulting image Z) during the second stage of processing.

The whole construction process is based on the processing precedents – a set of matched pairs of images $\{x|_{\theta(n_1, n_2)}, z(n_1, n_2)\}_{(n_1, n_2): \theta(n_1, n_2) \subseteq \Theta}$ (which are usually called *training dataset*) in order to minimize the processing error:

$$\varepsilon = \frac{1}{|\Theta|} \sum_{(n_1, n_2)} \|z - \Phi_2(\Phi_1(x))\| \rightarrow \min_{\Phi_1, \Phi_2}, \quad (1)$$

where Θ – the image domain, $\theta(n_1, n_2) \subseteq \Theta$ – restriction to the local fragment size $M_1 \times M_2$:

$$\theta(n_1, n_2) = \{(n_1 + m_1, n_2 + m_2) : m_1 = \overline{0, M_1 - 1}, m_2 = \overline{0, M_2 - 1}\}.$$

Characteristics of the decision

The most known solution of described task is based on the usage of artificial neural networks. Such approach has some special features, advantages and disadvantages which are well described in detail [4]. The alternative technology of the processing procedure construction is based on special hierarchical computational structures, such as *regression trees* and *decision trees* [5, 6]. These trees are the hierarchical structures consisting of 2 types of vertices - non-terminal vertices which define a partition of features domain, and terminal vertices which store a regression function.

The procedure based on regression trees has some advantages in comparison with the neural networks:

- automatic correction of "architecture" of the transformation;
- automatic selection of local features which is result of the partition process;

$\theta(n_1, n_2) \subseteq \Theta$, is an image fragment:

$$\theta(n_1, n_2) = \{(n_1 + m_1, n_2 + m_2) : m_1 = \overline{0, M_1 - 1}, m_2 = \overline{0, M_2 - 1}\}$$

(image constriction to region $\theta: x|_{\theta(n_1, n_2)}$),

$$\tilde{\theta}(n_1, n_2) = \{(n_1 + m_1, n_2 + m_2) :$$

$$m_1 = \overline{-(M_1/2), M_1/2}, m_2 = \overline{-(M_2/2), M_2/2}\}$$

is a centered fragment,

N_1, N_2 are the image dimensions, M_1, M_2 are the fragment dimensions (processing «window»).

The main task in the first stage is the formation features (some specific set of image properties) for predetermined local image fragment – $\bar{y} = (y_0, y_1, \dots, y_{K-1})^T$, $\bar{y} \in \mathbf{R}^K$ on the base of transformation $\Phi_1: \mathbf{R}^{M_1 \times M_2} \rightarrow \mathbf{R}^K$.

- finitely of the building and tuning process (computational efficiency);
- ease of tuning of the regression parameters in the terminal vertex.

There are some restrictions for the practical implementation. At first, the most important task on the stage of image features calculation in a "sliding window" mode is the task of development of a computationally efficient algorithm for this calculation. Moreover, this algorithm should allow consistently increase the features number up to whole system, because the traditional algorithms of the defining of the effective features subset based on iterative methods and computationally inefficient. At second, the main task on the stage of designing of hierarchical regression is the development of an algorithm to automatic construction of processing procedures on the base of the training dataset which be able to avoid the retraining and insufficient training problems.

Choosing of the linear local features types

We used the family of signal characteristics on the base of local wavelet discrete transformations (DWT) of the signals and images as an image features set. Such features have the following characteristics:

- existence of the computationally efficient calculation algorithm [9];
- complete description of the input signal;
- consistent obtaining and usage of features removes the problem of iterates on the features set.

Issues related to the features formation on the base of local DWT algorithms, as well as their advantages and speciality in relation to local image processing tasks, are considered in the work [9].

The classic scheme for fast calculating of local wavelet transformation (FWT) is based on Mallat scheme [4] and in accordance to the theory of a multiple-scale analysis can be represented as following equations:

$$w_{l+1}^+(p) = \sum_{n \in D_h} h(n-2p) \cdot w_l^+(n),$$

$$w_{l+1}^-(p) = \sum_{n \in D_g} g(n-2p) \cdot w_l^+(n),$$

where $p = \overline{1, N}$, N – length of the input signal, $h(n)$, $g(n)$ – such filters that: $\sum_n h(n) = 1$, $g(n) = (-1)^n h(-n+2t-1)$, ($t \in \mathbb{Z}$), D_h, D_g – sizes of domain areas, $l = \overline{0, \log_2 M}$ – wavelets levels, M – processing window size.

Concerning the image processing, the computational complexity of such algorithm for the wavelet levels $[L_1, L_2]$ can be evaluated [9] as following:

FWT on the base of Mallat scheme:

$$U_1^*(L_1, L_2) = 8/3(2^{2L_2} - 1);$$

Modified FWT: $U_2^*(L_1, L_2) \approx 8L_2 - 5L_1 - 5$;

Recursive FWT: $U_3^*(L_1, L_2) = 13(L_2 - L_1 + 1)$.

Regression tree construction

Technology of regression trees construction consists of the following stages:

1) Selecting parameters and method for hierarchical structure construction.

Here we need to distinguish vertices, which require a partition based on the error evaluation, as well as to determine parameters of vertex partition (a threshold and the number of vertices divided), while at the same time selecting the best feature, which cumulatively provide a maximum error reduction.

When using a linear regression:

$$f_j(\bar{y}) = \sum_{k=0}^{K-1} a_{jk} y_k + a_{jK} + \varepsilon, \quad (y_k \in [y_{k,j}^{\min}, y_{k,j}^{\max}]), \quad (2)$$

a processing error (1) of a terminal vertex can be represented as:

$$\varepsilon_j(k, \alpha) = \min_{\alpha} \left\| f_j^*(\bar{y}) - g(\bar{y}) \right\|_{[y_{k,j}^{\min}, y_{k,j}^{\alpha}]} + \min_{\alpha} \left\| f_j^{**}(\bar{y}) - g(\bar{y}) \right\|_{[y_{k,j}^{\alpha}, y_{k,j}^{\max}]}$$

where α^* is an optimal partition threshold,

k_j^* is the best feature.

2) Calculating the regression coefficients for each terminal vertex.

The elementary regression adjustment (construction) for a vertex is a regression coefficient calculation based on Ordinary Least Squares (a solution of system of linear algebraic equations for all vertex elements of the training set):

$$f(y) = \bar{a}^T \bar{y}; \quad \bar{y} = (y_0, y_1, \dots, y_{K-1}, 1)^T.$$

$$\varepsilon^2 = E \|\bar{a}^T \bar{y} - g(\bar{y})\|^2, \quad \tilde{\varepsilon}^2 = (\bar{a}^T U - \bar{\Gamma})^T (\bar{a}^T U - \bar{\Gamma}),$$

where $\bar{a} = (UU^T)^{-1}U\bar{\Gamma}$.

At the same time, in case of using the linear regression, it is possible, that the number of vertex elements in a set is insufficient to calculate the regression coeffi-

cients. There are different ways to solve this problem, such as: a rejection from the regression construction for the vertex, a coefficient "descent" from the upper level vertex, a reduction of the number of features considered for the vertex, etc.

However, the most effective solution is to conduct a regularization, i.e. to extend the definition of system of linear algebraic equations via the elements of the upper level set, and to solve the system with limitations, which guarantee a zero error for target terminal vertex points.

$$\begin{cases} \bar{a}^T \bar{y}_n = g(\bar{y}_n), & n = \overline{1, N}, \\ \left(\bar{a}^T U - \bar{\Gamma} \right)^T \left(\bar{a}^T U - \bar{\Gamma} \right) \rightarrow \min_{\bar{a}}. \end{cases}$$

To solve this problem we use the method of Lagrange multipliers based on the functional of the following form:

$$F = \sum_{m=1}^M \left(g(\bar{y}^{(m)}) - \sum_{k=0}^K a_k y_k^{(m)} \right)^2 + \sum_{n=1}^N \mu_n \left(\sum_{k=0}^K a_k y_k^{(n)} - g(\bar{y}^{(n)}) \right) \xrightarrow{\bar{\mu}, \bar{a}} \min.$$

Solutions of the respective system of equations are:

$$\begin{cases} \frac{\partial F}{\partial a_i} = \sum_{m=1}^M 2 y_i^{(m)} \left(g(\bar{y}^{(m)}) - \sum_{k=0}^K a_k y_k^{(m)} \right) - \sum_{n=1}^N y_i^{(n)} \mu_n = 0, \\ i = \overline{0, K}, \\ \frac{\partial F}{\partial \mu_j} = g(\bar{y}^{(j)}) - \sum_{k=0}^K a_k y_k^{(j)} = 0, \quad j = \overline{1, N}. \end{cases}$$

For the transformation of the second type (outputs of the form $\mathbf{K} = \{0, 1, 2, \dots, L-1\}$ – solution of classification problems), the solution is formed as:

$$f(y) = c, \quad \text{where } c = \left\{ l : n_l^v = \max_{j=0, L-1} n_j^v \right\}.$$

Here the analogous regularization consists in the use of "upper" level vertex classification results to determine the class number for target terminal vertex:

$$l = \arg \max_{j \in S_v} n_j^{v-1},$$

$$\text{where } S_v = \left\{ l : n_l^v = \max_{j=0, L-1} n_j^v \right\},$$

v is a terminal vertex,

v^{-1} is a non-terminal vertex, preceding the vertex v ,

S_v is a set of indices for the vertex v ,

n_j^v – number of j^{th} class objects in area of vertex v .

Here the classification error is:

$$\varepsilon = \frac{1}{N^2} \sum I(f(y), z), \quad \text{where } I(a, b) = \begin{cases} 0, & a = b \\ 1, & a \neq b \end{cases}$$

3) Checking the limitations on the computational complexity and evaluating the quality of the processing procedure for the control set.

Finishing of the construction process

To determine the stop parameters for algorithm construction process, we need to estimate the generalization

capability of the local processing procedures. When the amount of available data is limited, we cannot infinitely increase the decision rule complexity, otherwise the processing procedure will "maximally adapt" to the training set and will demonstrate bad results for other images of the class under consideration. On the other hand, if the procedure is "half-taught", the transformation error will be unacceptable for both training and control sets. Obviously, for every problem there is an optimal model complexity, which provides the best possible quality of generalization.

Statistical approach

Suppose, there is a set of objects for training:

$$\Omega = \{ \bar{y}_j, g(y_j) \}, j = \overline{1, T}.$$

To evaluate the quality of the algorithm, it is necessary to divide the set into "training" and "control" parts:

$$\Omega^T = \Omega^s \cup \Omega^t,$$

where $s + t = T, \Omega^s \cap \Omega^t = \emptyset$.

The algorithm quality for the set Ω is described by the expression:

$$v(a, \Omega) = (1 / |\Omega|) \sum_{\omega_i \in \Omega} I(\omega_i, a(\omega_i)), \quad (3)$$

where A is a family of algorithms,

a is a regression or classification algorithm (computable function),

μ is a method (constructed algorithm) for training via the set: $\mu(\Omega) = a$,

$$I(\omega_i, a(\omega_i)) = \begin{cases} 1, & |a(\omega_i) - \Phi(\omega_i)| \geq \delta(\omega_i) \\ 0, & |a(\omega_i) - \Phi(\omega_i)| < \delta(\omega_i) \end{cases} \quad \text{-- for regression construction problems,}$$

$$I(\omega_i, a(\omega_i)) = \begin{cases} 1, & a(\omega_i) \neq \Phi(\omega_i) \\ 0, & a(\omega_i) = \Phi(\omega_i) \end{cases} \quad \text{-- for classification problems.}$$

problems.

Therefore, according to [7], a functional of a uniform error rate deviation for two sets can be represented as:

$$P_\epsilon^{st}(A) = P \left\{ \sup_{a \in A} (v(a, \Omega^t) - v(a, \Omega^s)) > \epsilon \right\}. \quad (4)$$

It allows us to write the following limitation (for $s=t$):

$$P_\epsilon^{st}(A) \leq \Delta^A(2s) \cdot 1.5e^{-\epsilon^2 s},$$

where $\Delta^A(s)$ is a growth function for the family of algorithms, $\Delta^A(s) \leq 1.5(s^h / h!)$.

The use of statistical theory for risk estimation, in case of solving the regression construction problem, is as follows. Let $\bar{\alpha}$ be a vector of regression coefficients, and $\epsilon^2(\bar{\alpha})$ be a transformation error for a training set of size s . Then, under fairly general assumptions on the form of feature vector distribution, with a probability of $(1-\eta)$ for all regression functions of a family $A(\Omega, \bar{\alpha})$ it is appropriate to use estimation of the following form:

$$I(\bar{\alpha}) \leq \epsilon(\bar{\alpha}) / (1 - \sqrt{\frac{h \cdot (\ln(s/h) + 1) - \ln \eta}{s}}) = J(\bar{\alpha}), \quad (5)$$

where h is a capacity of decision function class, $I(\bar{\alpha})$ is an average risk (generally speaking, its minimization is the purpose of solving the regression construction problem).

In this case, if we specify the structure of training methods $\mu_1 \subset \mu_2 \subset \dots \subset \mu_M = \mu$ for the allowable family of algorithms $A(\Omega, \bar{\alpha})$, then it becomes possible to minimize the average risk functional by structure elements. For each constructed algorithm $a_p = \mu_p(\Omega, \bar{\alpha})$ we calculate the estimation $\bar{\alpha}_p$ using Ordinary Least Squares, then we search for the best estimation among $\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_M$ in sense of a minimum of expression $J(\bar{\alpha})$ from (5):

$$\bar{\alpha}_{opt} = \arg \min_{1 \leq \bar{\alpha} \leq M} (J(\bar{\alpha})),$$

which is the solution of regression construction problem with parameters $\bar{\alpha}_{opt}$.

In case of hierarchical regression with a linear function of elementary regression, for terminal vertices the structure $\mu_1 \subset \mu_2 \subset \dots \subset \mu_M = \mu$ produces "trees" nested within one another, which store the elementary regression parameters in each vertex. Here, $\bar{\alpha}$ regarded as a generalized vector of coefficients, which consists of regression coefficients for each terminal vertex of the tree.

The capacity of decision function class determines "the number of degrees of freedom" (the number of class parameters). This value is indirectly related to the amount of memory required to store hierarchical regression parameters. It is easy to see that for linear regression $h = K + 1$, for piecewise-linear regression $h = (K + 1)P$, for piecewise-constant regression $h = P + 1$, where K is a number of features, P is a number of terminal vertices of the tree.

Furthermore, according to [1, 2], we can write the following expression:

$$\eta < 4,5((2s)^{h-1} / (h-1)!)e^{-0,5\epsilon^2(s-1)}, \quad (6)$$

which characterizes the relationship between 3 parameters: ϵ – accuracy, η – reliability (trustworthiness) and s – size of a training set.

The considered method can be used to develop a stopping rule for construction process, but the basic problem of a statistical theory is overestimation due to excessive generalization. The theory is valid for any target function, arbitrary distribution of objects in space, and a very broad class of training methods, i.e. many significant features of the training process are not taken into account.

Combinatorial approach

A combinatorial approach arose as an attempt to construct a statistical learning theory more precisely, starting with its initial postulates [7]. It turned out that the algorithm model complexity is not a determining factor of training quality [8]. The way, the algorithm parameters are adjusted via set, is much more important. Combinatorial theory allows to substantiate the use of arbitrarily complex algorithm models, provided that they are configured appropriately. Major studies in combinatorial theory development are aimed at showing how the parameter adjustment should be performed to avoid the risk of retraining.

As mentioned above, low error rate for a given training set, in general, does not mean, that the constructed algorithm will also perform well for other sets. In this case, the error rate for a control set Ω' , which, in general, does not intersect with the training set Ω^s , also cannot adequately describe the training quality. The disadvantage of this approach is that, generally speaking, we fix a random set partition $\Omega^T = \Omega^s \cup \Omega'$ into training and control parts and even if the value $\varepsilon_{\Omega'}^2$ is small enough, there is no guarantee that, in case of the other partition $\Omega^T = \Omega^{s'} \cup \Omega'$ of the same set, the value $\varepsilon_{\Omega'}^2$ is also small.

From these considerations follows the requirement for the functional, which characterizes the training quality of the final set: it must be invariant under arbitrary set permutations [8].

Let (Ω_n^s, Ω_n^t) , $n = 1, 2, \dots, N$ be the all possible partitions of a set Ω^T into training and control sets. Let $v(\mu(\Omega_n^s), \Omega_n^t)$ be the error rate of an algorithm $\mu(\Omega_n^s)$, constructed on the basis of a set Ω_n^s , tested with the use of a set Ω_n^t (an analogue of function (3) for various combinations of control and training sets). The number of all N set partitions is C_T^s .

Full cross-validation functional, which characterizes the quality of $\mu(\Omega)$ method training for a finite set of objects Ω and demonstrates the invariance property:

$$Q_c^{st}(\mu(\Omega), \Omega) = \frac{1}{N} \sum_{n=1}^N v(\mu(\Omega_n^s), \Omega_n^t). \quad (7)$$

It is proved in [2], that the expected value of this quality functional is limited:

$$EQ_{v,\varepsilon}^{st}(\mu(\Omega), \Omega) < P_\varepsilon^{st}(A),$$

where $P_\varepsilon^{st}(A)$ is a functional of a uniform bias of an error rate for two sets (4) from Vapnik theory [7].

Next, if we consider h as a capacity of decision function class, it becomes possible to write an estimate for (7) in the following form:

$$Q_{v,\varepsilon}^{st}(\mu(\Omega), \Omega) < (C_T^0 + C_T^1 + \dots + C_T^h)(C_{T-\varepsilon}^s / C_T^s). \quad (8)$$

This estimate allows to significantly (by orders) reduce the requirements for the size of the training set. Empirical studies show, that this model selection technique is more preferable in many cases than principles of structural risk minimization and minimum description length, based on various formalizations of an algorithm complexity concept. In [2] the concept of effective capacity is introduced, and it is shown that statistical estimates remain correct, if we replace capacity with effective capacity. At the same time, in particular problems the effective capacity can be considerably lower than the total capacity of the family, e.g. in case of linear decision rules.

The effective capacity:

$$Q_{sup}^{st}(\mu, \Omega^T) = (1/N) \times \sum_{n=1}^N [\sup(v(\mu, \Omega_n^t) - v(\mu, \Omega_n^s)) > \varepsilon] < C(T^h e^{-\varepsilon^2 s} / h!),$$

where C is a constant, and $s = t$.

Moreover, we can estimate a local effective capacity – a value of a parameter h from the expression (8), such that the dependence $Q_{v,\varepsilon}^{st}(\mu(\Omega), \Omega)$ can be approximated by the following formula in the best way possible:

$$Q_{v,\varepsilon}^{st}(\mu(\Omega), \Omega) \approx C \frac{L^h}{h!} \max_{m \in M(\varepsilon, \sigma)} \sum_{k \in K(\varepsilon, \sigma)} \frac{C_m^k C_{T-m}^{s-k}}{C_T^s},$$

where $M(\varepsilon, \sigma) = \{m | \varepsilon t < m \leq t + \sigma s\}$,

$K(\varepsilon, \sigma) = \{k | \max(0, m - t) \leq k \leq \min(\sigma s, (m - \varepsilon t)s / T)\}$.

Unlike Vapnik concept of the effective capacity, the local effective capacity considers all three factors: features of the object distribution, features of the target function, and features of the training method.

Algorithm for automatic construction of processing procedure with limitations on complexity and execution quality

Development of effective algorithm of the local image processing based on a hierarchical regression and such features as a local DWT of image requires simultaneous consideration of different performance indicators: the computational complexity of the procedure, its quality or processing error and generalization capability.

The scheme of the algorithm of automatic construction of processing procedures is shown in Fig. 2. The algorithm assumes the consistent accumulation of features as long as the functional of a full sliding control is decreasing (that means that a quality of processing is improving), and the computational complexity of the procedure remains in predetermined limits.

Experimental researches

As an experimental task, we consider the task of image filtration. The solution involves the use of local image processing procedures based on regression tree (RT) and artificial neural network (NN). The comparison of the processing quality and computational complexity of these algorithms is presented in table 1.

Table 1. The comparison results

NN	ε	10.92	10.78	10.69	10.67	10.63	10.62	10.62	10.64
	U	54	99	189	279	369	459	549	621
RT	ε	11.28	11.01	10.89	10.81	10.72	10.62	10.57	10.61
	U	31	38	41	44	46	48	50	52

As can be seen from the table, the proposed method of hierarchical regression has the better accuracy with essentially smaller computational complexity than well known neural network method.

The experimental results allow us to draw the following conclusions:

- the constructed computational procedure for local processing demonstrates significantly higher efficiency (both speed and processing quality) for solving filtering and image restoration problems, compared with known Wiener filter;
- the computational procedure for local processing and its construction method demonstrate superiority over the method of processing and construction, based on the use of artificial neural networks. As for computational complexity, it is superior by orders.

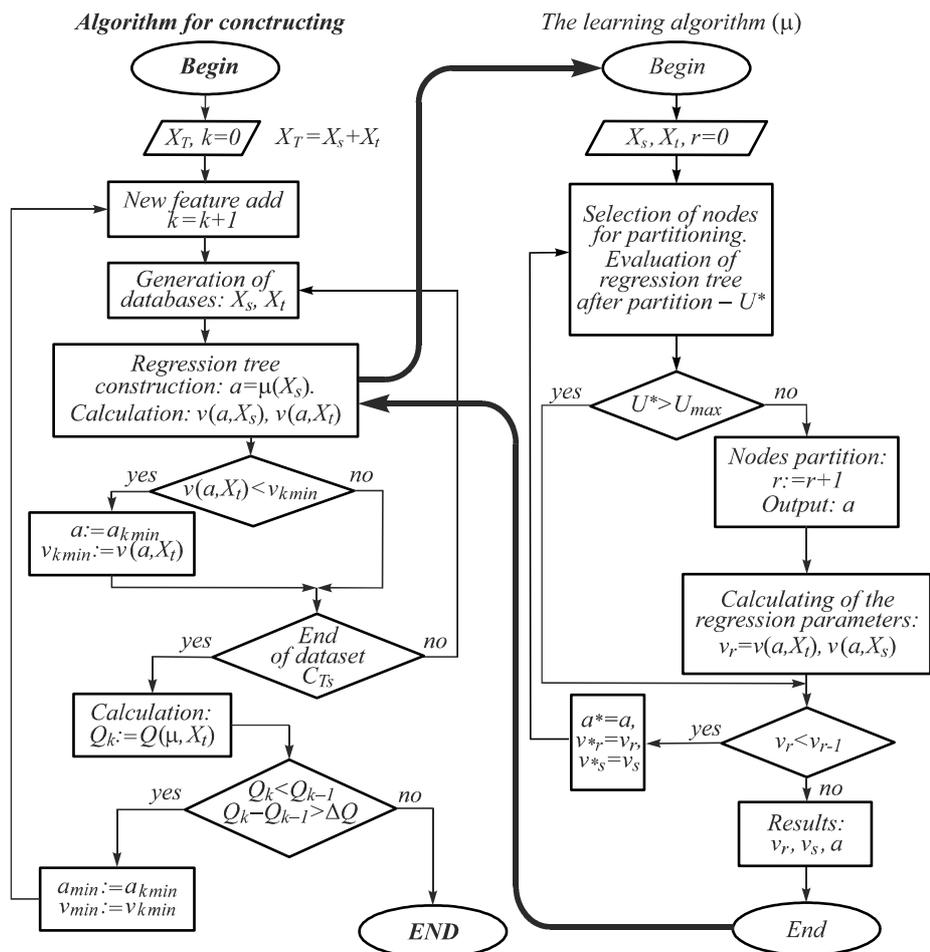


Fig. 2. A diagram of algorithm for construction a computational procedure of local image processing

Methodology of stopping of training process

Taking into account generalization capability of the local processing procedures based on a functional of a full sliding control [8] we can estimate that the total number of all possible N decompositions of dataset is C_T^s . General scheme of construction procedure is shown in Fig. 3.

Is quite logical fact that in the case of images processing the construction of processing procedures which takes into account all combination of training and testing dataset is unrealizable because of the incredibly large busting on various combinations of datasets. Therefore, we have had to develop a method of determination rule for stop busting process on the base of a finite number of samples.

For sufficiently large sample volumes can be assumed that the error rate of the algorithm has a binomial distribution with t degrees of freedom (the length of the test dataset) and the probability of "success" = p (quality of the algorithm on a control set).

In this way:

$$v(\mu(\Omega_n^s), \Omega_n^t) \sim Bin(t, p).$$

Function of the probability is specified as:

$$p_v(r) = C_t^r p^r (1-p)^{t-r}, \quad r = \overline{0, t}.$$

Then the distribution of the functional full cross-validation is evaluated by:

$$Q^s(\mu(\Omega), \Omega) = \frac{1}{N} \sum_{n=0}^{N-1} v(\mu(\Omega_n^s), \Omega_n^t) \sim Bin(N \cdot t, p).$$

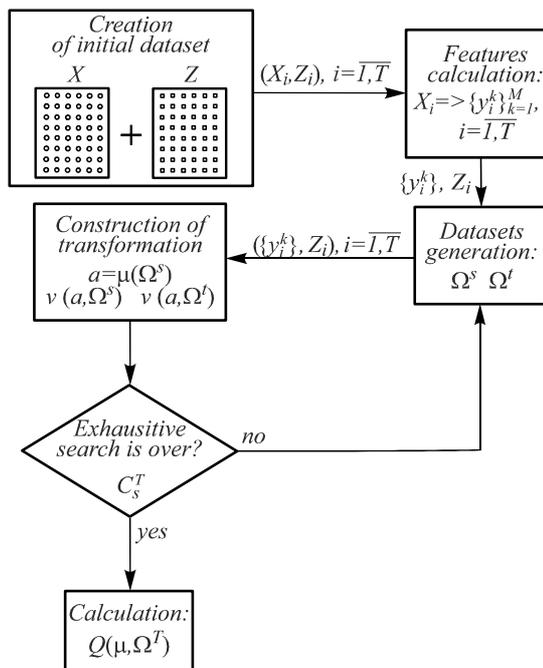


Fig. 3. Schema of construction procedure with exhaustive search on datasets

Decision about continuing or stopping generation of different combinations training and control datasets and transition to the next subset of features can be taken based on the analysis of functional $Q_1^{st} \sim Bin(N_1 \cdot t, p_1)$, $Q_2^{st} \sim Bin(N_2 \cdot t, p_2)$ for the different subsets of features. We decide whether to recalculate the feature space, or to stop the process of building a processing procedure under the assumption of $p_2 < p_1$, with veracity γ (and correspondingly $p_2 < p_1$, with the veracity $(1 - \gamma)$).

In such case the quality of the algorithm on dataset Ω can be estimated as:

$$v(\mu(\Omega), \Omega^T) \sim \frac{1}{|\Omega|} \sum_{\omega_i \in \Omega} I(\omega_i, \mu(\omega_i)),$$

where $I(\omega_i, \mu(\omega_i)) = \begin{cases} 1, & p \\ 0, & 1 - p \end{cases}$.

Moreover, if $n \gg 1$ (that is justified, because n is the number of objects and corresponds to the image size) and the λ is fixed, we obtain the Poisson distribution with parameter λ $Bin(n, \lambda/n) \approx P(\lambda)$.

In this case to make a decision of stop generation process for various combinations of training and control datasets, and to transition to next features set we have to calculate confidence intervals for the expectation of a Poisson distribution for the functional full cross-validation on a datasets N_1 and N_2 in form:

$$\left[\lambda_1 - (\tau_{1-\alpha/2} \sqrt{\lambda_1 / \sqrt{N_1}}), \lambda_1 + (\tau_{1-\alpha/2} \sqrt{\lambda_1 / \sqrt{N_1}}) \right] \times \left[\lambda_2 - (\tau_{1-\alpha/2} \sqrt{\lambda_2 / \sqrt{N_2}}), \lambda_2 + (\tau_{1-\alpha/2} \sqrt{\lambda_2 / \sqrt{N_2}}) \right],$$

where $\tau_{1-\alpha/2}$ – quantile of distribution $N_{0,1}$ for level $1 - \alpha/2$ ($\alpha = 1 - \gamma$).



Fig. 4. Calculation of confidence intervals

The decision of stopping generation of different combinations training and control datasets and about the switching to the next subset of features is taken at a moment when a separation of calculated confidence intervals on adjacent steps is achieved.

Illustration of the process of processing algorithm construction

Figure 5 shows an example of training of a regression tree, for different sets of features ($K=1, 2, 3, \dots, 12$, with a gradual increase). The graphs show the noise reduction (ϵ^2/D_v) with the increasing of a regression tree depth (H_{cp}). Fig. 6 presents a statistics of process of regression tree construction on a various combinations of training and control datasets (group of points of each colors correspond to the optimum value of quality for a given set of features $K=1, 2, 3, \dots, 12$, in the case of exhaustive search a some number of partitioning of a dataset Ω on training and control part (Ω_n^s, Ω_n^t), $n = 1, 2, \dots, N$).

Fig. 7 shows a graph of the construction process of local image processing procedures, with confidence intervals, for the optimal values of quality, and Figure 8 – calculation of the required number of combinations of training/testing datasets for the making a decision of switching to the next set of features (the number of combinations required for the separation of the confidence intervals on the adjacent steps).

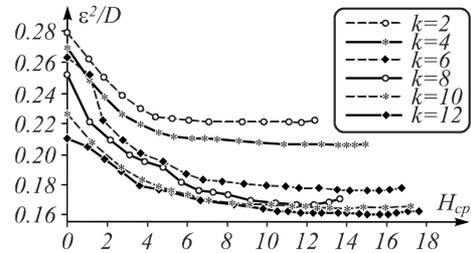


Fig. 5. Process of training of processing procedure at various features space

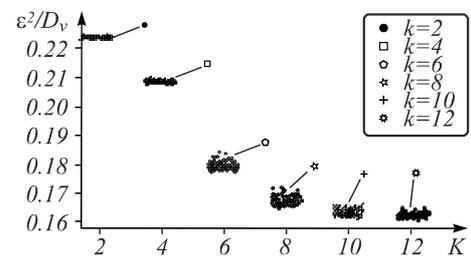


Fig. 6. Statistics of quality of procedures for different combinations of training and control datasets

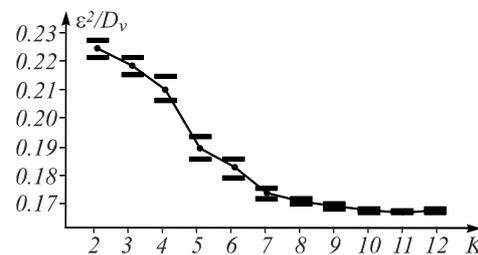


Fig. 7. Construction of the processing procedure with confidence intervals

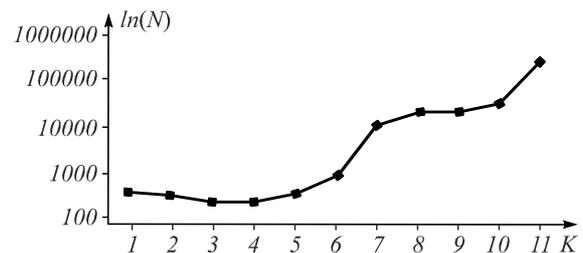


Fig. 8. Calculating of combinations number

Conclusions and results

The paper presents an *efficient technology* that allows to realize automatic construction of computational procedure of local processing of digital signals/images. In accordance with the creation processes, the constructed computational procedure has a specified complexity, the highest quality, and the generalizing ability. The proposed method of estimation of the required number of algorithm training iterations and, as a consequence, the

stopping rule of the formation different combinations of training and testing datasets based on their particular number allows to use the full functionality of combinatorial theory and a functional of full cross-validation control during the constructing (training) processing procedures, which are tuning on the bases of a training dataset. And as a result, possible to prevent problems of retraining/poorly trained processing algorithms and, at the same time, construct the local processing procedure with predetermined computational complexity and veracity, and with the best quality (for an existing training dataset).

References

- [1] Soifer VA, ed, Chernov AV, Chernov VM, Chicheva MA, Fursov VA, Gashnikov MV, Glumov NI, Ilyasova NY, Khramov AG, Korepanov AO, Kupriyanov AV, Myasnikov EV, Myasnikov VV, Popov SB, Sergeyev VV. Computer Image Processing, Part II: Methods and algorithms. VDM Verlag; 2010. ISBN: 978-3-639-17545-5.
- [2] Woods R, Gonzalez R. Digital Image Processing. 2nd ed. Prentice Hall; 2002. ISBN 0-201-18075-8.
- [3] Pratt W. Digital image processing. 4th ed. Wiley-Interscience; 2007. ISBN: 978-0-471-76777-0.
- [4] Haykin S. Neural Networks: A Comprehensive Foundation. Upper Saddle River, NJ: Prentice Hall; 1999. ISBN 0-13-273350-1.
- [5] Breiman L, Friedman J, Olshen R, Stone C. Classification and regression trees. Monterey, CA: Wadsworth, Inc.; 1984. ISBN 978-0412048418.
- [6] Kopenkov V, Myasnikov V. An algorithm for automatic construction of computational procedure of non-linear local image processing on the base of hierarchical regression [In Russian]. Computer optics 2012; 36(2):257-266.
- [7] Vapnik, VN, Chervonenkis AY. Theory of Pattern Recognition [in Russian]. Moscow: "Nauka" Publisher; 1974.
- [8] Vorontsov K. A combinatorial approach to assessing the quality of training algorithm [In Russian]. Mathematical problems of cybernetics 2004; 13: 5-36.
- [9] Kopenkov V. Efficient algorithms of local discrete wavelet transform with HAAR-like bases. Pattern Recognition and Image Analysis 2008; 18(4): 654-661. DOI: 10.1134/S1054661808040184.
- [10] Kopenkov V. On halting the process of hierarchical regression construction when implementing computational procedures for local image processing. Pattern Recognition and Image Analysis 2014; 24(4): 506-510. DOI: 10.1134/S1054661814040087.

Authors' information

Vasiliy Nikolaevich Kopenkov (1978 b.), graduated from S.P. Korolyov Samara State Aerospace University (SSAU) at 2001, received his Candidate's degree in Technical Sciences in 2011. At present he is an assistant at SSAU's Geoinformatics and Information Security sub-department, holding a part-time position of a researcher at the Image Processing Systems Institute of the Russian Academy of Sciences. The area of research interests includes digital signals and image processing, geoinformatics, pattern recognition. He is a co-author of 53 scientific papers, including 10 research articles. He is a member of the Russian Association of Pattern Recognition and Image Analysis. E-mail: vkop@smr.ru . Website: <http://www.ipsi.smr.ru/staff/kopenkov.htm> .

The information about author **Vladislav Valerievich Myasnikov** you can find on page 712 this issue.

Code of State Categories Scientific and Technical Information (in Russian – GRNTI):27.43.51

Received September 26, 2016. The final version – October 19, 2016.