Dynamic Scale Adaptation Algorithm of Image Etalon Functions

Mikhail Gavrikov and Roman Sinetsky
Department of Software Engineering, Platov South-Russian State Polytechnic University, 132 Prosvecshenia Str., Novocherkassk, Russia
gmm1000@yandex.ru, rmsin@srspu.ru

Keywords: Image Recognition, Pattern Recognition, Prototype Function, Scale-Adapted Function.

Abstract: An algorithm for large-scale adaptation of prototype functions representing image classes is proposed. The algorithm identifies the parameters of nonlinear scale distortions contained in the functions representing the observed image realizations, and then transforms the original prototype function using the previously proposed model. The algorithm works on a class of images of regular phase processes that have the property of quasi-similarity of shape. Two models of large-scale nonlinear transformations are considered: symmetric and asymmetric. The differences between the models and the practical results of their application are given. The algorithm was experimentally tested on the images of prototypes of fragments of speech signals, electrocardiosignals, and engine cylinder pressure detector signals. Examples and experimental data confirming the effectiveness of the algorithm are given. Conclusions are formulated about the possibility of using the algorithm with both models in practical problems.

1 INTRODUCTION

Most image recognition methods for various physical processes use the operation of comparing the current image with the prototype (etalon) image. For example, speech recognition algorithms use many prototype spectral images of phonemes, algorithms for processing data of technical and medical diagnostics can use prototypes of signals or spectral functions characterizing changes in the parameters of controlled physical processes in various modes and states (normal operation, pathology of certain types, etc.). If the features of images belonging to the same class change little over time (within small limits relative to statistical averages), then the prototype images remain unchanged during the analysis. If the specified parameters can dynamically change within large limits during the processing of the registered image implementations, then for the reliable operation of the recognition algorithm, the parameters of the prototype images must adapt to these changes. That is, the prototype images must be dynamic.

In applied problems, prototype images are often represented in the form of real functions: fragments of signals of finite duration, analytically defined functions approximating these fragments, power spectral densities, etc. In the case of dynamic images, if the image of the prototype is represented by a function $y_0(x)$, and recorded on the interval T ($T \subset X$, X is the real axis) current image of the function $\tilde{y}_T(x)$, then before comparison it is necessary to perform the transformation $y_0(x) \to y_T(x)$ prototype functions $y_0(x)$ in some new function $y_T(x)$ that has the meaning of the adapted prototype with which to compare a function $\tilde{y}_T(x)$ that represents the current image.

Among the physical processes that are the sources of the analysed images, a significant place is occupied by the class of *regular phase processes* (RF-processes) [1, 2, 3]. In particular, such processes include dynamic processes occurring in internal combustion engines, electrical processes of polarization and depolarization of the heart muscle, etc. A characteristic feature of the class of images of RF processes is that all registered implementations of image functions $\tilde{y}_T(x)$ from this class have the property of "similarity of form" (resemblance, but not coincidence). In [4], such images were called "pulsating" and two models for their formation were proposed. In general, these models can be represented in the form of a converter $H(\tilde{s}_k, \tilde{s}_a)$ that performs large-scale transformations of the

prototype $y_0(x)$ into simulated implementations of $\hat{y}(x)$:

$$\hat{y}(x) = H(\tilde{s}_k, \tilde{s}_a)[y_0(x)], \quad (1)$$

where $\tilde{s}_k : x \to kx$, $\tilde{s}_a : y_0 \to ay_0$ is the scale transformations with random parameters k, a. If the parameters of the scale transformations are not constants, but are themselves functions of the argument $x : k = k(x)$, $a = a(x)$, then the scale distortions are nonlinear, and the functions $\tilde{y}_T(x)$ generated according to (1) will have the specified shape similarity property.

The aim of this work is to develop and experimentally test an algorithm designed to identify the parameters of nonlinear scale distortions contained in the observed implementations of a class of functions, and to construct an adapted prototype function representing this class of images.

2 THE ADAPTATION TASK

Let $e(\tilde{y}_T(x), y_0(x))$ be the deviation calculated in some way between the observed function $\tilde{y}_T(x)$ and the prototype $y_0(x)$. The task of adaptation is set as follows:

Having an observable function $\tilde{y}_T(x)$, a prototype function $y_0(x)$, and a model of scale transformations $H(\tilde{s}_k, \tilde{s}_a)$, we need to obtain estimates of the transformation parameters k, a for which the function $\hat{y}(x) = H(\tilde{s}_k, \tilde{s}_a)[y_0(x)]$ satisfies the condition

$$e(\tilde{y}_T(x), \hat{y}(x)) < e(\tilde{y}_T(x), y_0(x)), \quad (2)$$

where $e(\tilde{y}_T(x), \hat{y}(x))$ is the deviation between the corresponding functions.

The model function $\hat{y}(x)$ that satisfies condition (2) can be used as an adapted prototype function $y_T(x) = \hat{y}(x)$, with which the implementation of $\tilde{y}_T(x)$ representing the current image should be compared. A possible modification of such a statement of the problem may consist in replacing condition (2) with condition

$$e(\tilde{y}_T(x), \hat{y}(x)) < e^{\partial},$$

where e^{∂} is the specified value of the permissible deviation. Obviously, in both cases, the problem may have many solutions or not have a solution, and the problem statement itself contains a conceptual scheme for obtaining a solution, if any.

This scheme involves the following steps:

- processing of implementation $\tilde{y}_T(x)$, as a result of which estimates of the parameters of scale transformations $\tilde{s}_k, \tilde{s}_a$ should be obtained;
- substitution of the obtained parameter values into the used model of the converter $H(\tilde{s}_k, \tilde{s}_a)$ and generation of the adapted prototype function $\hat{y}(x) = H(\tilde{s}_k, \tilde{s}_a)[y_0(x)]$;
- checking condition (2) and, if it is fulfilled, replacing the original prototype function $y_0(x)$ with an adapted prototype function $y_T(x) = \hat{y}(x)$.

3 SCALE CONVERTER MODEL

As a model of the transformer $H(\tilde{s}_k, \tilde{s}_a)$, we can use a modification of the relations obtained in [4] to model a set of implementations of $y_n(x)$ by stochastic nonlinear scale distortions of the prototype function $y_0(x)$. In this work, the relations given below do not contain the index *n*, since they are used not for modelling, but for generating an adapted function when performing the second stage of the above conceptual adaptation scheme.

The prototype function is given in the form of a piecewise function

$$y_0(x) = \begin{cases} y_0^{(1)}(x), & x_{(0)}^{\bullet} = 0 \le x < x_{(1)}^{\bullet}, \\ \ldots \\ y_0^{(m)}(x), & x_{(m-1)}^{\bullet} \le x \le x_{(m)}^{\bullet} = b^{\bullet}, \end{cases}$$

defined by some partition

$$\pi_m(D^{\bullet}) = \{d_i \mid d_i = [x_{(i-1)}^{\bullet}, x_{(i)}^{\bullet}),\ i = 1, \ldots, m,\ d_m = [x_{(m-1)}^{\bullet}, x_{(m)}^{\bullet}],$$
$$0 = x_{(0)}^{\bullet} < x_{(1)}^{\bullet} < \ldots < x_{(m)}^{\bullet} = b^{\bullet}\}$$

of the domain of the function definition $D^{\bullet} = [0, b^{\bullet}]$ with nodal points $q(x_{(i)}^{\bullet}, y_{(i)}^{\bullet})$ [5, 6].

Modelling of $\hat{y}(x)$ is performed by linear scale transformations $y_0^{(i)}(x) \to \hat{y}^{(i)}(x)$ of each i-th segment of the piecewise prototype function, but with different values of the scale transformation parameters for different segments.

The functions $\hat{y}^{(i)}(x)$ formed at the output of the converter are determined by the relations:

$$\hat{y}^{(i)}(x) = a^{(i)} y_0^{(i)}\left(\frac{x - x^{\bullet}_{(i-1)}(1-b^{(i)}) - \delta^{(i-1)}}{r\ b^{(i)}}\right) + \\ + y_0(x^{\bullet}_{(i-1)})(1-a^{(i)}) + \xi^{(i)}, \quad (3)$$

where $a^{(i)}, b^{(i)}$ is the coefficients of scale transformations:

$$a^{(i)} = 1 + \frac{\xi^{(i)} - \xi^{(i-1)}}{y_0(x^{\bullet}_{(i)}) - y_0(x^{\bullet}_{(i-1)})},\ b^{(i)} = 1 + \frac{\delta^{(i)} - \delta^{(i-1)}}{x^{\bullet}_{(i)} - x^{\bullet}_{(i-1)}},$$

that depend on the values of random parameters $\xi^{(i)},\ \delta^{(i)}$ with the characteristics:

$$M\{\xi^{(i)}\} = 0,\ \xi^{(i)} \in [-\Delta_y^{(i)}, \Delta_y^{(i)}],$$
$$M\{\delta^{(i)}\} = 0,\ \delta^{(i)} \in [-\Delta_x^{(i)}, \Delta_x^{(i)}],$$

$r = const$ is the coefficient of linear change in the scale of argument x over the entire interval $D^{\bullet} = [0, b^{\bullet}]$.

The use of a constant coefficient r leads to linear compression/expansion of the prototype function $y_0(x)$, and the use of different values of the coefficients of scale transformations $a^{(i)}, b^{(i)}$ for its different segments leads to nonlinearity of scale distortions that will be contained in the resulting function $\hat{y}(x),\ x \in T = [0, \hat{b}],\ \hat{b} = rb^{\bullet}$.

Internal node points $q(x_{(i)}, y_{(i)})$ of the modelled piecewise function (with the exception of the boundary points $q(x_{(0)}, y(x_{(0)}))$, $q(x_{(m)}, y(x_{(m)}))$) satisfy the constraint:

$$(x_{(i)}, y_{(i)}) \in \Theta_{(i)} = [x^{\bullet}_{(i)} - \Delta_x^{(i)},\ x^{\bullet}_{(i)} + \Delta_x^{(i)}] \times \\ \times [y_0(x^{\bullet}_{(i)}) - \Delta_y^{(i)},\ y_0(x^{\bullet}_{(i)}) + \Delta_y^{(i)}] \quad (4)$$

which means that they fall into the *rectangles of movement of nodal points* $\Theta_{(i)}$.

This feature binds the contours of the simulated functions $\hat{y}(x)$ to the contour of the prototype $y_0(x)$ and preserves the similarity property of their shapes. The values of $\Delta_x^{(i)},\ \Delta_y^{(i)}$ set the allowable increments of the ranges of the definition areas and the ranges of values of the corresponding segments of the functions $y^{(i)}(x)$, provided that the adjacent rectangles $\Theta_{(i)}$ should not intersect: $\Theta_{(i)} \cap \Theta_{(i+1)} = \varnothing$. Therefore, $\Delta_x^{(i)},\ \Delta_y^{(i)}$ is defined as:

$$\Delta_x^{(i)} = c_x \min\left\{(x^{\bullet}_{(i)} - x^{\bullet}_{(i-1)}),\ (x^{\bullet}_{(i+1)} - x^{\bullet}_{(i)})\right\},$$
$$\Delta_y^{(i)} = c_y \min\left\{(y_0(x^{\bullet}_{(i)}) - y_0(x^{\bullet}_{(i-1)})),\right. \\ \left.(y_0(x^{\bullet}_{(i+1)}) - y_0(x^{\bullet}_{(i)}))\right\}, \quad (5)$$
$$0 \le c_x, c_y < 0.5.$$

The rectangles $\Theta_{(i)}$ defined in this way are symmetric with respect to the node points $q(x^{\bullet}_{(i)}, y^{\bullet}_{(i)})$ of the prototype $y_0(x)$.

A possible modification of this method is to set the rectangles that are not symmetric in the coordinate while maintaining the condition $\Theta_{(i)} \cap \Theta_{(i+1)} = \varnothing$:

$$\Theta_{(i)} = [x^{\bullet}_{(i)} - \overleftarrow{\Delta}_x^{(i)}, x^{\bullet}_{(i)} + \vec{\Delta}_x^{(i)}] \times \\ \times [y_0(x^{\bullet}_{(i)}) - \Delta_y^{(i)}, y_0(x^{\bullet}_{(i)}) + \Delta_y^{(i)}], \quad (6)$$
$$\overleftarrow{\Delta}_x^{(i)} = c_x(x^{\bullet}_{(i)} - x^{\bullet}_{(i-1)}),\ \vec{\Delta}_x^{(i)} = c_x(x^{\bullet}_{(i+1)} - x^{\bullet}_{(i)}).$$

From relation (3) it follows that for an unambiguous specification of the converter it is necessary to determine the value of r and the set of values $\Omega = \left\{(\xi_n^{(i)}, \delta_n^{(i)}),\ i = \overline{1, m-1}\right\}$, on which the parameters of scale transformations $a^{(i)}, b^{(i)}$ depend.

4 ALGORITHM DESCRIPTION

For modelling functions with a specific shape, it was suggested in [4] to choose as the nodal points of the prototype function not arbitrary points [7, 8], but essential points $q(x, y),\ y = y_0(x)$, where significant changes in the properties of the function occur. The set of formal features used to identify each essential point $q(x, y)$ (for example, the features of the extremums of a function) and determining the behaviour of the function $y_0(x)$ in its neighbourhood determines the type $type(q)$ of this point [1, 2]. The choice of the composition of the types of essential points and the method (algorithm) of their automatic identification depends on the form or other properties of the function and is carried out by the developer of the signal processing system based on knowledge of the subject. The set of node points and their function types $y_0(x)$ will be denoted as $Q^{\bullet}$ and $TYPE^{\bullet}$, respectively:

$$Q^{\bullet} = \left\{q(x^{\bullet}_{(i)}, y^{\bullet}_{(i)}),\ i = \overline{(1, m)}\right\},$$

$$TYPE^{\bullet} = \left\{type\ q(x^{\bullet}_{(i)}, y^{\bullet}_{(i)}),\ i = (\overline{1,m})\right\}.$$

The logic of the algorithm for identifying the parameters of scale distortions of functions given below is based on the following considerations.

The desired scale-adapted function $\hat{y}(x) = H(\tilde{s}_k, \tilde{s}_a)[y_0(x)]$ must have a geometric similarity of shape both with any of the implementations of $y_n(x)$ modelled using model (3), including the prototype function $y_0(x)$, and with the observed function $\tilde{y}_T(x)$. The characteristic features of the form of a function are largely determined by the quantitative relations between the values of the coordinates of its essential points, as well as the order in which their types are located. In order to ensure the specified similarity, the following conditions must be met:

- types of nodal points $\hat{q}(\hat{x}_{(i)}, \hat{y}_{(i)})$ of the function $\hat{y}(x)$ needs to match the types of the nodal points $q(x^{\bullet}_{(i)}, y^{\bullet}_{(i)})$ of the original function prototype $y_0(x)$, and the points to get into the rectangles of movement of nodal points $\Theta_{(i)}$:

$$type\ (\hat{x}_{(i)}, \hat{y}_{(i)}) = type\ q(x^{\bullet}_{(i)}, y^{\bullet}_{(i)}),$$
$$(\hat{x}_{(i)}, \hat{y}_{(i)}) \in \Theta_{(i)};$$

- the transformation $q(x^{\bullet}_{(i)}, y^{\bullet}_{(i)}) \to \hat{q}(\hat{x}_{(i)}, \hat{y}_{(i)})$ of the node points of the prototype function $y_0(x)$ to the node points (with new coordinates) of the function $\hat{y}(x)$ should be performed only if the corresponding rectangle $\Theta_{(i)}$ contains at least one essential point $q(x, \tilde{y})$ of the $type\ q(x^{\bullet}_{(i)}, y^{\bullet}_{(i)})$ of the observed function $\tilde{y}_T(x)$, that is, if the condition is met:

$$(x, \tilde{y}) \in \Theta_i,\quad type\ q(x, \tilde{y}) = type\ q(x^{\bullet}_{(i)}, y^{\bullet}_{(i)}).$$

In addition, the domain $D^{\bullet} = [0, b^{\bullet}]$ of the prototype $y_0(x)$ must be adjusted to the domain $T = [0, \tilde{b}]$ of the observed function $\tilde{y}_T(x)$. To do this, perform a linear transformation of the scale of the argument x and create a prototype:

$$y'_0(x) = y_0\left(\frac{x}{r}\right),\quad r = \tilde{b} / b^{\bullet}. \qquad (7)$$

The coordinates of the node points q'_i of the prototype $y'_0(x)$ will be $(rx^{\bullet}_{(i)}, y'_0\ (rx^{\bullet}_{(i)}))$. These conditions are taken into account in the algorithm when identifying scale distortions and constructing a scale-adapted function $\hat{y}(x)$.

Algorithm A1. Algorithm for dynamic scale adaptation of the function.

Input:

$y_0(x)$ – the prototype function;

$\tilde{y}_T(x)$ – the observed function;

$Q^{\bullet} = \left\{q(x^{\bullet}_{(i)}, y^{\bullet}_{(i)}),\ i = (\overline{1,m})\right\}$ – the set of nodal points for the function $y_0(x)$;

$TYPE^{\bullet} = \left\{type\ q(x^{\bullet}_{(i)}, y^{\bullet}_{(i)}),\ i = (\overline{1,m})\right\}$ – the set of types of nodal points for the function $y_0(x)$.

Output:

$y_T(x)$ – the scale-adapted prototype function.

Step 1. Using the relations (7), create a prototype $y'_0(x)$:

$$y'_0(x) = y_0\left(\frac{x}{r}\right),\ x \in T = [0, \tilde{b}],\quad r = \tilde{b} / b^{\bullet}$$

(The coordinates of the nodal points q'_i of the prototype $y'_0(x)$ will be $(rx^{\bullet}_{(i)}, y'_0\ (rx^{\bullet}_{(i)}))$);

Step 2. Define the set of essential points $\tilde{Q}$ of the observed function $\tilde{y}(x)$ as:

$$\tilde{Q} = \left\{q(x_{(j)}, \tilde{y}_{(j)}) \mid type\, q(x_{(j)}, \tilde{y}_{(j)}) \in TYPE^{\bullet},\ j = (\overline{1,n})\right\}$$

(to determine this set, we can use the algorithm for constructing a piecewise monotone function from [1]);

Step 3. Calculate the deviation $e(\tilde{y}(x), y'_0(x))$ between the observed function $\tilde{y}(x)$ and the prototype $y'_0(x)$:

$$e(\tilde{y}(x), y'_0(x)) = \int_0^b \left|\tilde{y}(x) - y'_0(x)\right| dx$$

Step 4. For each node point of the prototype function $y'_0(x)$, define symmetric:

$$\Theta_{(i)} = [rx^{\bullet}_{(i)} - \Delta^{(i)}_x,\ rx^{\bullet}_{(i)} + \Delta^{(i)}_x] \times$$
$$\times [y'_0\ (rx^{\bullet}_{(i)}) - \Delta^{(i)}_y,\ y'_0\ (rx^{\bullet}_{(i)}) + \Delta^{(i)}_y]$$

or non-symmetric (optional):

$$\Theta_{(i)} = [rx^{\bullet}_{(i)} - \vec{\Delta}^{(i)}_x,\ rx^{\bullet}_{(i)} + \vec{\Delta}^{(i)}_x] \times$$
$$\times [y'_0\ (rx^{\bullet}_{(i)}) - \Delta^{(i)}_y,\ y'_0\ (rx^{\bullet}_{(i)}) + \Delta^{(i)}_y]$$

rectangles, respectively (see relations (5-6));

Step 5. Generate sets C_i of essential points with *type* $q(x^{\bullet}_{(i)}, y^{\bullet}_{(i)})$ of the observed function $\tilde{y}(x)$ falling into rectangles $\Theta_{(i)}$:

$$C_i = \{ q(x_{(j)}, \tilde{y}_{(j)}) \mid q(x_{(j)}, \tilde{y}_{(j)}) \in \tilde{Q},$$
$$(x_{(j)}, \tilde{y}_{(j)}) \in \Theta_{(i)}, type\ \tilde{q}(x_{(j)}, y_{(j)}) = type\ q(x^{\bullet}_{(i)}, y^{\bullet}_{(i)}) \}$$

and determine the cardinalities N_i of these sets (C_i may be empty);

Step 6. Determine the coordinates of the node points $\hat{q}(\hat{x}_{(i)}, \hat{y}_{(i)})$ of the function $\hat{y}(x)$ as

$$(\hat{x}_{(i)}, \hat{y}_{(i)}) = \begin{cases} \left(\frac{1}{N_i} \sum_{j=1}^{\hat{N}_{(i)}} x_{(j)}, \frac{1}{N_i} \sum_{j=1}^{\hat{N}_{(i)}} \tilde{y}_{(j)} \right), \\ \qquad q(x_{(j)}, \tilde{y}_{(j)}) \in C_i, if\ C_i \neq \varnothing, \\ (x^{\bullet}_{(i)}, y^{\bullet}_{(i)}),\ \ if\ \ C_i = \varnothing, \end{cases}$$

and the set Ω of estimates of the scale distortion parameters $(\xi^{(i)}, \delta^{(i)})$ as

$$(\xi^{(i)}, \delta^{(i)}) = \begin{cases} (\hat{x}_{(i)} - r x^{\bullet}_{(i)},\ \hat{y}_{(i)} - y^{\bullet}_{(i)}),\ \ if\ C_i \neq \varnothing, \\ 0,\ if\ C_i = \varnothing; \end{cases}$$

Step 7. Using the relations (3) and the found values of the scale distortion parameters $(\xi^{(i)}, \delta^{(i)})$, construct a piecewise function $\hat{y}(x)$.

Step 8. Calculate deviation

$$e(\tilde{y}(x), \hat{y}(x)) = \int_0^b |\tilde{y}(x) - \hat{y}(x)|\, dx;$$

Step 9. Define a scale-adapted function

$$y_T(x) = \begin{cases} \hat{y}(x),\ \ if\ \ e(\tilde{y}(x), \hat{y}(x)) < e(\tilde{y}(x), y_0(x)), \\ y_0(x),\ \ if\ \ e(\tilde{y}(x), \hat{y}(x)) \geq e(\tilde{y}(x), y_0(x)). \end{cases}$$

End.

5 EXPERIMENTS

In the experimental testing of the algorithm, the prototype images of $y_0(x)$ were used:

a) spectral power density of fragments of speech signals corresponding to certain sounds of human speech (phonemes) [9, 10, 11, 12];

b) fragment of an electrocardiogram (ECG) that corresponds to several complete cycles of the heart at a certain heart rate, seismo- or gyro-cardiogram signals [13, 14, 15];

c) fragment of the function of pressure in the cylinder of a four-stroke internal combustion engine (ICE) versus time (the standards were obtained by analytical calculation at 8900 rpm) [16].

Two types of points were used as the nodal points of the prototype $y_0(x)$: local minima and maxima, which divide the function into intervals $\tilde{y}_T(x)$ of non-strict monotony, on which the function has a positive or negative trend [7]. As observed functions $\tilde{y}_T(x)$ with nonlinear scale distortions, we used implementations $y_n(x)$ generated using the model (3) with 10000 implementations for each case (a) and (b). In the algorithm, both methods of determining the rectangles of movement of the nodal points $\Theta_{(i)}$ were implemented: symmetric and asymmetric.

An illustration of the algorithm is shown in Figure 1, Figure 2 and Figure 2. In Figure 1, the broken line shows the prototype spectrum $y_0(x)$ for the phoneme "E", the dotted line shows the observed spectrum $\tilde{y}_T(x)$, the solid line shows the scale-adapted prototype $y_T(x) = \hat{y}(x)$, as well as the symmetrical rectangles of the movement of the nodal points $\Theta_{(i)}$. Similar ECG graphs are shown in Figure 2.

Figure 1 shows that the nodal points of the observed spectrum $\tilde{y}_T(x)$ fall into the second and third rectangles, so the average peak of the adapted spectrum $y_T(x)$ is pulled up to the average peak of the observed spectrum $\tilde{y}_T(x)$ and in this area differs significantly from the prototype $y_0(x)$.

From Figure 2 it can be seen that the nodal points of the observed ECG cycle fall into four rectangles and the scale adaptation is performed in the area corresponding to the QRS complex and the T wave.

In fig. 3, it can be seen that the key points corresponding to the maximum pressure value that is observed at the moment of ignition of the combustible mixture fall into the movement rectangles, and the standard is adapted for one-cylinder operation cycle.

In all cases, the deviation of the adapted prototype from the observed function is reduced, in particular, for the example shown in Figure 1, the deviation is $e(\tilde{y}_T(x), y_0(x)) = 133476$, $e(\tilde{y}_T(x), y_T(x)) = 119742$, which is 10% less and satisfies the condition (2).

Tables 1, 2 and 3 provide a summary of the results of experiments to evaluate the effectiveness of the algorithm. Two indicators were evaluated:

- *adaptation rate* expressed as the percentage of cases in which the algorithm performed prototype adaptation with a decrease in deviation, relative to the total number of experiments;
- the *relative decrease R of the deviation* $e(\tilde{y}_T(x), \hat{y}(x))$ compared to the deviation $e(\tilde{y}_T(x), y_0(x))$, calculated as

$$R = \frac{e(\tilde{y}_T(x), y_0(x)) - e(\tilde{y}_T(x), \hat{y}(x))}{e(\tilde{y}_T(x), y_0(x))} 100\% .$$

The minimum, maximum, and average *R* values of 10000 implementations are calculated separately.

These indicators are of interest, since they characterize the "degree of relevance" of applying the procedure for adapting etalon images (prototypes) in the problems of parametric identification and pattern recognition in specific

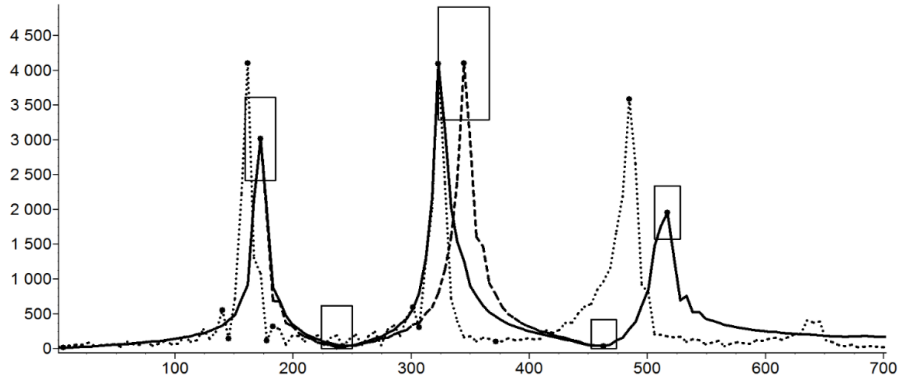

Figure 1: Illustration of the algorithm for the speech signal.

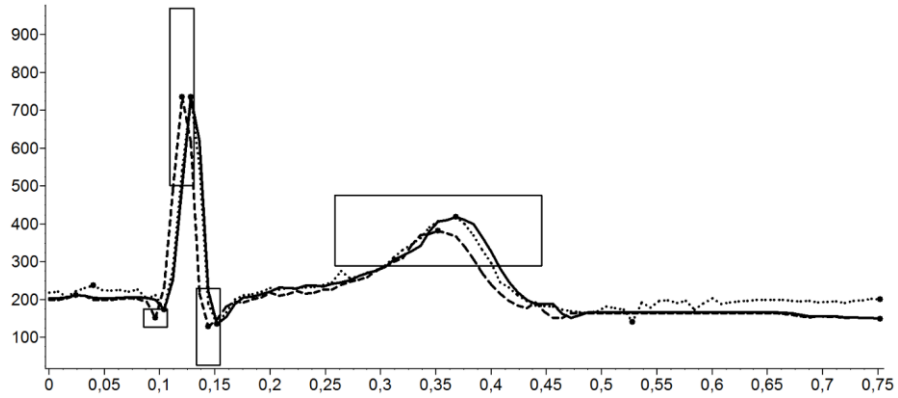

Figure 2: Illustration of the algorithm for the electro cardio signal.

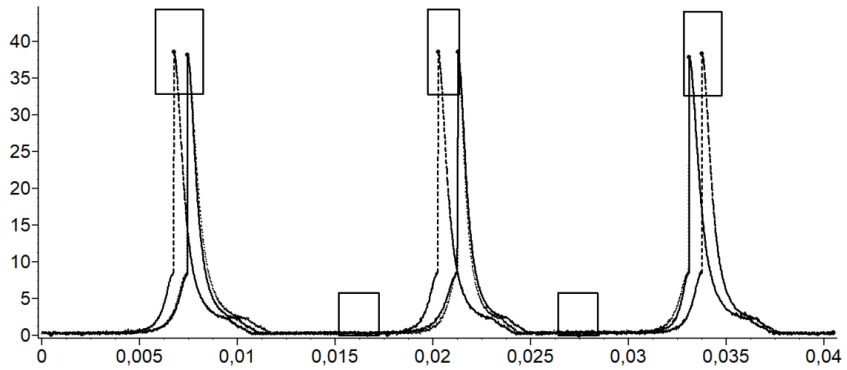

Figure 3: Illustration of the algorithm for cylinder pressure function.

applied areas. The higher the values of these indicators, the higher the "degree of relevance" of applying the adaptation procedure using the above algorithm. These indicators were calculated using the symmetric and asymmetric rectangles algorithm.

Table 1: Results for the speech signal.

Rectangle type	Adaptation rate, %	Relative deviation decrease R, %		
		Min	Average	Max
Symmetrical	65	0,02	5,06	14,24
Asymmetrical	91	0,23	7,95	21,52

Table 2: Results for the electro cardio signal.

Rectangle type	Adaptation rate, %	Relative deviation decrease R, %		
		Min	Average	Max
Symmetrical	94	0,17	29,21	58,95
Asymmetrical	97	0,32	38,06	71,85

Table 3: Results for the cylinder pressure function.

Rectangle type	Adaptation rate, %	Relative deviation decrease R, %		
		Min	Average	Max
Symmetrical	93	0,00	20,97	47,74
Asymmetrical	99	19,6	49,67	77,03

6 CONCLUSIONS

Two main conclusions follow from the analysis of the results obtained.

The frequency of adaptation of the prototype of more than 50% indicates the adequacy of the algorithm of the real situation, which was discussed at the beginning of this work. Static definition of an image class in the form of an immutable prototype function can lead to significant deviations of the functions representing the observed images from this class, due to their scale distortions, and to recognition errors, in which an image belonging to class $\{y_0(x)\}$ is recognized as not belonging to it. Using the prototype adaptation algorithm allows you to significantly reduce these deviations.

When using asymmetrical rectangles, the expected higher adaptation rates are achieved. At the same time, if the value of the constant c_x, which defines the asymmetrical rectangles in the relations (6), is not selected correctly (for example, close to 0.5), a different kind of error may occur, when the prototype $y_0(x)$ can be adapted to the image of another class. In this case, an image that does not belong to class $\{y_0(x)\}$ will be incorrectly recognized as belonging to it.

REFERENCES

[1] M. M. Gavrikov, "Structural approximation and recognition of one-dimensional time images", Concept and applications. Russian Electromechanics, 2003, vol 6, pp. 52-60.

[2] U. Grenander, "Lectures in Pattern Theory", Volume I-III, Springer, 1976-1981, Berlin.

[3] R. van der Vlist, C. Taal, and R. Heusdens, "Tracking Recurring Patterns in Time Series Using Dynamic Time Warping", 2019 27th European Signal Processing Conference (EUSIPCO), A Coruna, Spain, 2019, pp. 1-5, doi: 10.23919/EUSIPCO. 2019.8903102.

[4] M. M. Gavrikov and R. M. Sinetsky, "Algorithms for the simulation of signals and spectral functions with a pulsating scale distortions", University news, North-Caucasian region, Technical sciences series, 2013, vol 3, pp. 3-9.

[5] M. M. Gavrikov and R. M. Sinetsky, "Algorithms for segmentation of structural time images and their application in speech signal processing", University news, North-Caucasian region, Technical sciences series, 2010, vol 1, pp. 18-24.

[6] A. Stan, C. Valentini-Botinhao, B. Orza, and M. Giurgiu, "Blind speech segmentation using spectrogram image-based features and Mel cepstral coefficients", 2016 IEEE Spoken Language Technology Workshop (SLT), San Diego, CA, 2016, pp. 597-602, doi: 10.1109/SLT.2016.7846324.

[7] L. S. Fainzilberg, "Information technologies for processing complex waveforms", Theory and practice, Ukraine, Kiev: Naukova dumka, 2008, 336 p.

[8] A. Mazumdar and L. Wang, "Covering arbitrary point patterns", 2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 2012, pp. 2075-2080, doi: 10.1109/Allerton.2012.6483478.

[9] M. M. Gavrikov and R. M. Sinetsky, "Algorithmic and numerical implementation of the structural approximation method for speech pattern recognition", Russian Electromechanics, 2007, vol 2, pp. 52-59.

[10] J. B. Allen, "Short-term spectral analysis, synthesis, and modification by discrete Fourier transform", IEEE Trans. on Acoustics, Speech, Signal Processing, 1997, vol. ASSP-25. N 3, pp. 235-238.

[11] K. Vijayan and K. S. R. Murty, "Analysis of Phase Spectrum of Speech Signals Using Allpass Modeling", IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 12, pp. 2371-2383, Dec. 2015, doi: 10.1109/TASLP.2015.2479045.

[12] L. V. Zlatoustova, R. K. Potapova, and V. N. Trunin-Donskoy, "General and applied phonetics", Moscow, MSU, 1986, 304 p.

[13] L. S. Fainzilberg, “Heart functional state diagnostic using pattern recognition of phase space ECG-images”, Proc. 6th European Congress on Intelligent Techniques and Soft Computing (EUFIT '98). Aachen (Germany), 1998, N B-27, pp. 1878-1882.

[14] M. Yochum, Ch. Renaud, and S. Jacquir, “Automatic detection of P, QRS and T patterns in 12 leads ECG signal based on CWT”, Biomedical Signal Processing and Control, Elsevier, 2016, ff10.1016/j.bspc.2015.10.011ff. ffhal-01328478

[15] C. Yang, N. D. Aranoff, P. Green, and N. Tavassolian, “Classification of Aortic Stenosis Using Time–Frequency Features From Chest Cardio-Mechanical Signals”, IEEE Transactions on Biomedical Engineering, vol. 67, no. 6, pp. 1672-1683, June 2020, doi: 10.1109/TBME.2019.2942741.

[16] Q. Wang, T. Sun, Z. Lyu, and D. Gao, “A Virtual In-Cylinder Pressure Sensor Based on EKF and Frequency-Amplitude-Modulation Fourier-Series Method”, Sensors 2019, 19, 3122, [Online]. Available: https://doi.org/10.3390/s19143122.