Neural net algorithms of construction of adaptive grid

S. Arkhipov

Buriyad State University, Ulan-Ude, Russia

Abstract

The brief review of known neural net algorithms of construction of adaptive grids are presented here. Often used variants of functions of neurons adjacency are resulted. Dependence of quality of grid adaptation from function parameters has been researched on the basis of using Gaussian function about the distance between neurons in modified algorithm of network SOFM. According to computing experiments recommendations at the choice of function parameters of the adjacency at adaptation of flat grid on complex area by modified algorithm SOFM have been resulted here.

1 Introduction

The construction of adaptive grid is important issue because its necessary to get reliable numerical calculation of applied problems at moderate requirements to computer facilities. The main idea of methods of adaptive grid construction is to reduce the sizes of cells in zones calculated area appropriating to significant mistakes of the problem.

Traditional methods such as equidistribution [1], Thompson [2], elliptical method [3], algebraical methods [4], conformal mapping [5] have been used for construction of adaptive grids. The theories of differential equations, calculus of variations and multidimensional differential geometry lay down the foundations of these methods. For receiving qualitative adaptive grids all these methods require to solve complex systems of the nonlinear differential equations with partial derivatives that have some restrictions.

At present the neuron net algorithms [6, 7, 8, 9, 10] and etc. demonstrate an ability to construct adaptive grids with predetermined density on complex physical area. Thanks to T.Kohonens classical theory of self-organizing maps (SOFM - Self-Organizing Feature Maps, T.Kohonen, [11, 12]) modern neuron net models have been developing today. Competitive neuron net SOFM with training without teacher performs the problem of mapping of multidimensional space to space with reduced dimensionality for example two-dimensional space. Discrete-stochastic approach which is used in training of neuron nets provides the simplicity of algorithms, possibilities of its effective paralleling, density reflection of data distribution and absence of affixment to dimension of mirrored space.

It is known the application of basic model SOFM is the cause of boundary effect, presence of dead neurons and malfunction of grid smoothness. For the solution of these problems we need to apply modified methods which based on the idea of alternation of basic algorithms for internal and external nodes [7] and also to use so-called painted models and special algorithms of smoothing [9]. Besides the functions of adjacency neurons are upgraded in the algorithms. This function represents not increasing function from discrete time and distance between the best matching unit and adjacent neurons in the grid. The function of adjacency neurons consists of two parts: proper function of distance and function of training speed. Variation of parameters of distance functions and training speed concerning discrete time provides quality of grid adaptation on area. As a rule, reasonable adaptation of the grid comes as a result of numerous computing experiments at the choice of parameters of adjacency neuron functions. At the same time the recommendations concerning distance functions and training speed have enough general character and slightly facilitate search of optimum parameters.

An attempt of formulation of recommendations at the choice of parameters of adjacency neuron functions including the Gaussian function of distance for modified algorithm SOFM has been undertaken and the brief review of existing neuron net algorithms of adaptation regular grids on flat area has been presented in this work.

2 Concept

Self-organizing feature map has a set of input elements which corresponds to dimensionality of training vectors (space of physical area) and a set of output elements which are called cluster elements (grid nodes) and serve as prototypes. The input elements pass signals to cluster elements by means of weighted connections. Weight connections are interpreted as value of coordinates describing cluster position in space of samples. In an initial instant of discrete time cluster elements can be defined as random coordinates of space of samples and in the set tops of regular grid, for example, with triangular, square or hexagonal edges. During training the weight of neurons are adjusted as result of which the grid is self-organized and stretched on the set physical area.

The network SOFM is characterized by initialization stage of feature map and cycle:

1. choice of random sample $x(n)$ with the set density of distribution;
2. finding the best matching unit (BMU) cluster on feature map which weight has smaller difference in the set metrics from the random sample;
3. updating of units from among nearby to the best matching unit variation of weight of the best matching unit and its neighbors for approximation to the random sample;
4. definition of feature map mistake.

As a rule Euclidean distance d is selected as the metrics for determining of the best matching unit in the step 2. The angle between radius vectors of applicant unit and random sample is used in rare cases.

The updating of units in step 3 occurs with adjacency function $\Theta(n, i_{BMU}, j_{BMU}, i, j)$ depending on proximity to the best matching unit under the formula:

$$w_{ij}(n+1) = w_{ij} + \Theta(n, i_{BMU}, j_{BMU}, i, j)(x(n) - w_{ij}(n)), \tag{1}$$

where n - number of iteration, w_{ij} - weight of ij unit, $x(n)$ - incidentally chosen sample, $i_{BMU}j_{BMU}$ - index of the best matching unit for the sample $x(n)$.

Function of adjacency represents not increasing function from discrete time n and distances between the best matching unit and adjacent neurons in the grid. As noted above, this function is broken on two parts: function of distance $h(d,n)$ and speed training function $\delta(n)$ i.e. $\Theta(n, i_{BMU}, j_{BMU}, i, j) = h(d,n) \cdot \delta(n)$.

Usually one of two distance functions can be used:

$$h(d,n) = \begin{cases} const, & d \le \sigma(n) \\ 0, & d > \sigma(n) \end{cases}$$

- step function or Gaussian function

$$h(d,n) = e^{-\frac{d^2}{2\sigma^2(n)}}. \tag{2}$$

At adaptation of grid on complex area the Gaussian function shows the best result. The function $\sigma(n)$ is known as radius of training which is selected sufficiently large at initial stage of training and decreased so only best matching unit is trained. As radius of training the linearly or exponentially decreasing function from time has been used, for example, in work [8]

$$h(d,n) = a \cdot n^{-0.2}, \tag{3}$$

where a is selected to receive perceptible displacement of all units of feature map. The grid distance $d^2 = (i_{BMU} - i)^2 + (j_{BMU} - j)^2$ is offered to use as distance d in the formula (2).

Function of training speed $\delta(n)$ also represents the decreasing function from time. It is often used linear, inversely proportional function n

$$\delta(n) = \frac{const_1}{n + const_2},$$

or

$$\delta(n) = n^{-0.2}. \tag{4}$$

In work [9] functions of training speed $\delta(n)$ and distance $h(d,n)$ are modified to form

$$\delta(n) = n^{-0.2} \cdot \left(1 - e^{\frac{5(n-n_{max})}{n_{max}}}\right),$$
$$h(d,n) = s^{-\frac{d^2}{r^2(n)}},$$

where n_{max} - maximal number of iterations, s - constant close to zero, d - Euclidean distance between neurons, $r(n)$ - radius of training

$$r(n) = r_{min} + \left(1 - e^{\frac{5(n-n_{max})}{n_{max}}}\right)\left(r_{max} \cdot s^{\frac{n}{n_{max}}} - r_{min}\right) \cdot n^{-0.25},$$

r_{min}, r_{max} initial and final radiuses of training.

As we can see from reduced functions all of them contain parameters which are result of matching. And it considerably influences to the quality of adaptive grid construction that means high heuristics of algorithm.

We can call above described algorithm as basic and using its with different variants of adjacency function which leads to three main problems:

1. Grid adaptation on nonconvex area G does not guarantee that all units of grid will belong to area G.
2. Boundary units of the constructed grid are located on the certain distance up to border of the area, distinct from zero. This distance is comparable to an average of distances between units of grid.
3. Disfunction of smoothness of adaptive grid because of shortening of training radius at specification stage.

For solving the two first problems in the work [7] which is devoted to using SOFM for construction finite-element grids the idea of modification of training algorithm SOFM has been offered. The idea of modification means alternation of using of this algorithm separately for boundary and internal units. One cycle of such alternation is called macroiteration [8]. According to this idea modified algorithm of construction of finite- difference adaptive grids has been developed [9].

The modified algorithm.

1. Initialization of positions of grid units.
2. On the first macroiteration ($s = 1$) the basic algorithm during n_0 iterations to all units of grid is applied.
3. On each macroiteration with number ($s > 1$) following actions are carried out:
   (a) Application of basic algorithm during $n_1(s)$ iterations to boundary units of grid with generation of point only on border of area.
   (b) Application of basic algorithm during $n_2(s)$ iterations to all units with generation of point in all area. In this case all boundary units are fixed and do not change the position. Besides if the best matching unit is the boundary one it can replace random point $x(n)$.
4. Macroiterations are repeated until variation of positions of units do not become small enough.

Solving of the third problem is to use boundary neurons as representatives of nonexistent neurons outside feature map, which do not suffice for balancing of non-central lateral connections. The algorithm realizing a smoothing of grid is applied after application of the modified algorithm and is beyond the given review.

From the given review it is clear, that known modified algorithm of grid adaptation has inherited heuristics of basic algorithm. The practice of application of modified method points out of necessity of availability of recommendations at the choice of training parameters.

3 Calculations and supervision

Computing experiments have been carried out on symmetric nonconvex area G (fig. 1). The choice of this area G is explained by evidence of a proper arrangement of grid at the first step of the modified algorithm. As function of the adjacency neurons $\Theta(n, i_{BMU}, j_{BMU}, i, j)$ the product of functions of training speed (4) and distances (2) has been applied. The modified algorithm of training is software-based algorithm in the form of following steps.

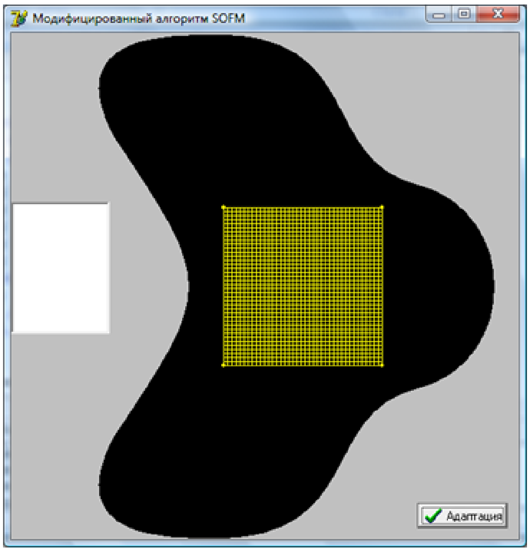


Fig. 1: Area G with an initial arrangement of grid in the adaptation field

1. Initial weights w_{ij} of all neurons in tops of a square grid are established (fig. 1).
2. It is carried out following actions on the first macroiteration ($s = 1$) corresponding to discrete time $n \in [1, n_0]$:
   (a) The random point $x(n)$ in all area G is generated;
   (b) The best matching neuron in Euclidean metric is defined. Grid coordinates of the best matching neuron i_{BMU}, j_{BMU} are fixed;
   (c) New weight values of grid are adjusted from the formula (1)

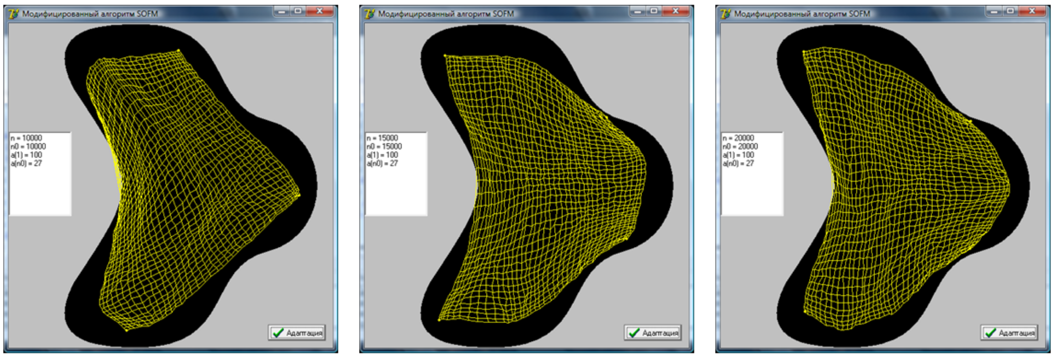
$$\Theta(n, i_{BMU}, j_{BMU}, i, j) = n^{-0.2} \cdot e^{-\frac{(i_{BMU}-i)^2+(j_{BMU}-j)^2}{2\cdot(a(n)\cdot n^{-0.2})^2}}. \tag{5}$$

3. It is carried out on each macroiteration ($s > 1$)
   (a) During $n_1(s)$ iterations
      i. The random point $(x(n))$ on border of area (G) is generated;
      ii. The best matching neuron BMU from boundary units of a grid is defined;
      iii. New weight 5 values of boundary units of grid are adjusted.
   (b) During $n_2(s)$ iterations
      i. The random point $x(n)$ in all area G is generated;
      ii. The best matching neuron BMU among all units of grid is defined;
      iii. If boundary neuron of grid wins in the step 3(b)ii randomly generated point $x(n)$ will be replaced onto boundary neuron;
      iv. New weight (5) values of internal units of a grid are adjusted.

4. Macroiterations are repeated until variation of positions of units does not become small enough.

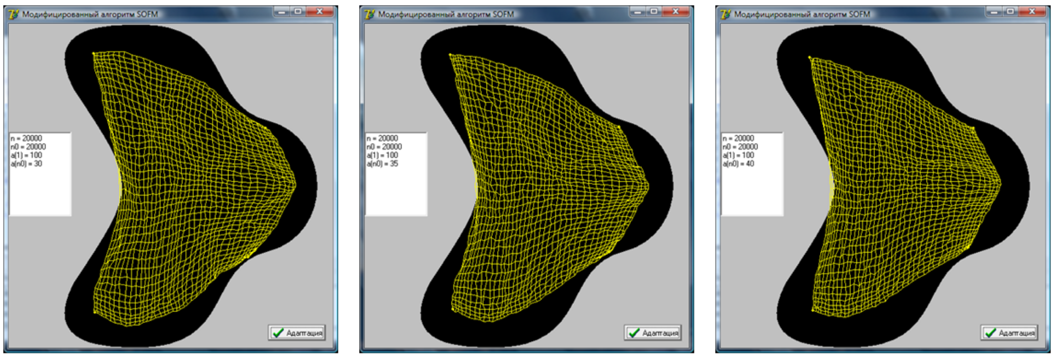
The first macroiteration corresponds to stage of ordering in classical terminology. The important feature is correctness of preliminary grid position in the area here. The subsequent macroiterations specify grid position concerning the border and internal area G.

It is necessary to divide stages of ordering and specification for the analysis of quality of grid construction in the modified algorithm. In the following computing experiments parameters a and n_0 have been researched. For a basis linear dependence of parameter from discrete time has been accepted at the first stage of training.



a) b) c)

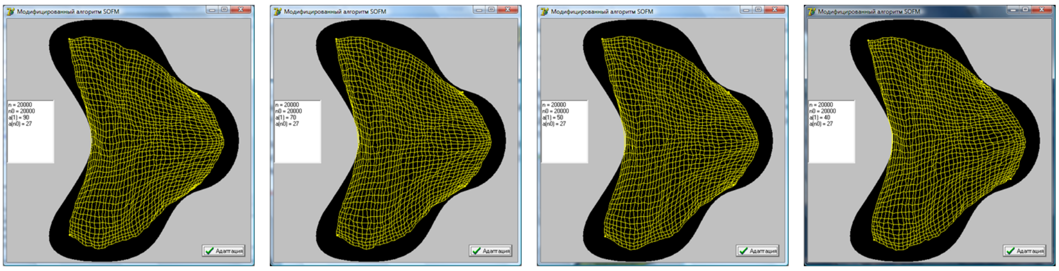
Fig. 2: Preliminary arrangement of grid in the field of at $a(1) = 100$, $a(n_0) = 27$ after n_0 iterations: a) $n_0 = 10000$ b) $n_0 = 15000$ c) $n_0 = 20000$



a) b) c)

Fig. 3: Preliminary arrangement of grid at the field of after 20000 iterations for different versions of parameter a) $a(n_0) = 30$ b) $a(n_0) = 35$ c) $a(n_0) = 40$

The increase of the bottom limit of parameter at constant n_0 and $a(1)$ leads to degradation of quality of preliminary grid construction (fig. 3). It can be explained by increase of the bottom limit of training radius $\sigma(n)$ (3) grid units are subjected to displacement at final stage and adaptive properties of grid has been become worse. The figure 4 illustrates the dependence of quality of construction from upper limit of parameter at constant parameters n_0 and $a(n_0)$. As we can see from the figure 4c decrease of upper limit leads to decreasing of adaptation quality of grid. This identifies the grid has been shrunk to incidentally generated points, decreasing in the sizes in the first iterations at the big radius of initial training. Then it is slowly unrolled adapted for features of area. The grid omits the compression stage at decrease of initial radius and initial sizes of the grid reduce possibility of correct adaptation.



a) b) c) d)

Fig. 4: Preliminary arrangement of grid at the field of after 20000 iterations for different versions of parameter a: a) $a(1) = 90$ b) $a(1) = 70$ c) $a(1) = 50$ d) $a(1) = 40$

Character of parameter decreasing from initial value $a(1)$ to $a(n_0)$ has been researched (fig. 5). Computing experiments with using exponential decreasing functions have shown (fig. 6) experiments the character of function decrease $a(n)$ has not brought essential deviations as construction of preliminary grid.

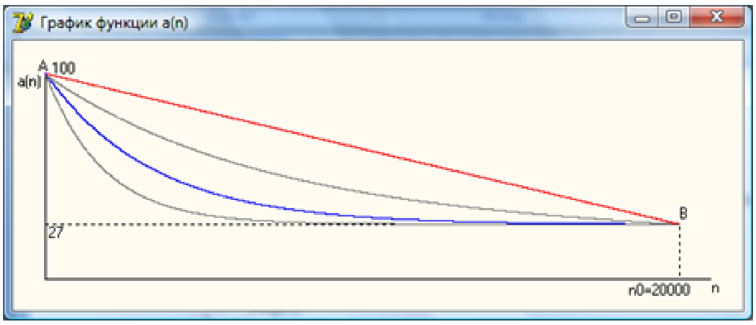
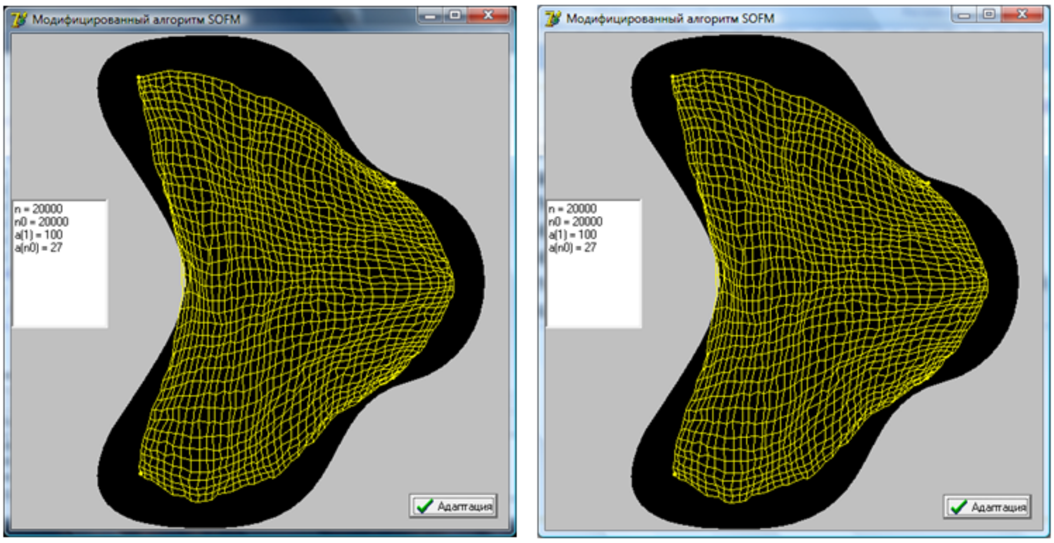


Fig. 5: Graph of decreasing functions $a(n)$ from the point A to B

Thus on stage of ordering the quality of preliminary construction of grid has been provided with correct task of points $A(1, a(1))$ and $B(n_0, a(n_0))$ in area of function construction $a(n)$. The function $a(n)$ can decrease linearly or exponentially.

On stages of specification, experiments have detected expediency of using of exponential decreasing function $a(n)$ from B to some point $C(n_{max}, a(n_{max}))$, disposed closely to axis of



a) b)

Fig. 6: Preliminary construction of the grid for a) exponentially decreasing function $a(n)$ b) linearly decreasing function $a(n)$

discrete time n. Use of linearly decreasing function from B to C keeps quality of adaptation, but it significantly increases time of calculations.

Example of construction of adaptive grid for function $a(n)$

$$a(n) = \begin{cases} \frac{a(n_0)-a(1)}{n_0} \cdot (n_0 - n) + a(1), & 1 \le n \le n_0 \\ a(n_0) \cdot \left(1 - e^{\frac{5(n-n_{max}-n_0)}{n_{max}+n_0}}\right) \cdot (0.005)^{\frac{n-n_0}{n_{max}-n_0}} + a_{min}, & n_0 < n \le n_{max} \end{cases} \tag{6}$$

is represented on (fig. 7). Fig. 8 illustrates graph of function a(n).

4 Conclusion

Existing neuron net algorithms prove possibility of construction adaptive grids on complex physical areas. The result of grid adaptation depends on heuristics of choice of parameters of neuron net training. It is more appropriate as parameter in Gaussian function of distance to use linearly decreasing function on the first macroiteration and exponentially decreasing function for receiving the best result in modified algorithm SOFM. The preliminary construction of grid has been provided by correct task of points $A(1, a(1))$ and $B(n_0, a(n_0))$ at plane of function construction $a(n)$.

References

[1] Hakimzyanov G., Shokin Y., Barahnin V., Shokina N. Numerical simulation of fluid flow with surface waves // Novosibirsk: Publishing house of the Siberian Branch of the

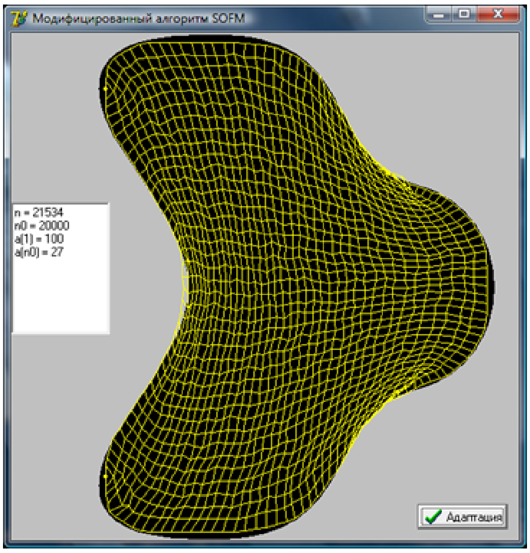


Fig. 7: Result of construction of adaptive grid with application (6)

Russian Academy of Science, 2001. 394 p.

[2] Thompson J.F., Warsi Z.U.A., Mastin C.W. Numerical grid generation, foundations and applications // Amsterdam: North-Holland, 1985.

[3] Liseykin V., Lebedev A., Kitaeva I. Universal elliptic method of construction of computational grids // Novosibirsk: NSU, 2004. 266 p.

[4] Gordon W.J., Thiel L.C. Transfinite mappings and their applications to grid generation // Numerical Grid Generation, Appl. Mathematics and Computation, vol. 2/3, 1982. pp. 171-192.

[5] Godunov S., Prokonov G. About calculations of conformal images and construction of computational grids // The magazine omputational mathematics and mathematical physics, vol. 7, 1967. pp.1031-1059.

[6] Ritter H., Martinetz T., Schulten K. Neural Computation and Self-Organizing Maps: An Introduction. New York: Addison-Wesley, 1992.

[7] Manevitz L., Yousef M. Finite-Element Mesh Generation Using Self-Organizing Neural Networks // Microcomputers in Civil Engineering 12, 1997. pp. 233-250.

[8] Nechaeva O. Neuron network approach for construction adaptive grids // Neuroinformatics, p. 2, 2006. pp.172-179.

[9] Nechaeva O. Compositional algorithm for construction of adaptive grids of random structure // The Collection of proceedings of All-Russia scientific and technical conference Neuroinformatics -2007, Moscow Engineering and Physical Institute, 2007. pp. 72-79

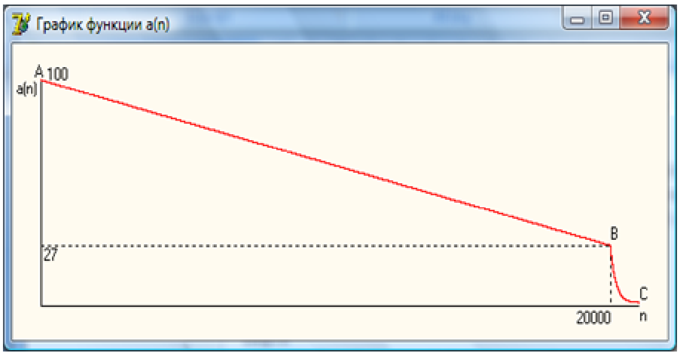


Fig. 8: The function graph $a(n)$ described by proportion (6)

[10] Koutnik J., Mazl R., Kulich M.. Building of 3d environment models for mobile robotics using self-organization // In Proc, of The 9th International Conference on Parallel Problem Solving From Nature - PPSN-IX, Springer, 2006. pp. 721-730.

[11] Kohonen . Self-organizing Maps // Springer Series in Information Sciences, V.30, Springer, Berlin, Heidelberg, New York, 2001. 501 p.

[12] Kohonen T.K. Self-organization and associative memory. New York: Springer Verlag, 1989. 312 p.