# Evolutionary Optimization of Growing Neural Gas Parameters for Object Categorization and Recognition

Guillermo S. Donatti, Oliver Lomp, Rolf P. Würtz, *Senior member, IEEE*

***Abstract*— The already introduced Neural Map provides a structural association for the building blocks of dynamically generated object models. Its learning and recall procedures are built upon the Growing Neural Gas algorithm, which is highly parameterized. The values of these parameters are obtained through a time-consuming empirical approach. In the present work, we evaluate the use of optimization based on Evolutionary Algorithms to simplify this task. This paradigm delves into six different approaches given by the combination of three fitness functions and two starting conditions. The performance of the proposed optimization paradigm is cross-validated with experiments on invariant object categorization and recognition found in literature. The results show that the empirically set parameter values can be successfully optimized using this paradigm.**

## I. Introduction

Organic Computing attempts to create systems that self-organize. This self-organization must be constrained by environmental data and the desires of the designer. These constraints can be fulfilled, but usually at the expense of introducing new parameters to the system. In the present work, we try to reduce the number of parameters of an object categorization and recognition system by leaving the optimal choice to an evolutionary algorithm. We show that the resulting system is successful, even on a task for which the parameters are not optimized.

A well-known class of algorithms for the self-organization of data has been developed on the basis of Kohonen's self-organizing map [1], a neural network model motivated by how visual, auditory, and other sensory information is handled by separate parts of the cerebral cortex in the human brain [2]. It maps high-dimensional vectors onto a graph with a given topology.

An extension, the Neural Gas [3], also subjects the graph topology to self-organization, leaving only the number of graph nodes to be determined beforehand. The Growing Neural Gas (GNG) algorithm [4], [5] goes one step further and learns the topology of the graph from sample data. This algorithm sequentially evaluates the samples' information and locally adapts their approximated topology based on a distance measure. Unfortunately, each of these extensions increases the number of parameters of the algorithm. There are a number of successful applications of the GNG [6], [7], [8] but the parameter tuning for a new problem remains awkward.

The Organic Computing approach to computer vision requires self-organized learning and processing of image features. Following this approach we have proposed the Neural Map [9], a memory framework for visual object categorization and recognition. It is composed of a GNG network [4] and a classifier motivated by the population coding and decoding processes of cortical neurons [10]. Its properties include the unsupervised structural association of elementary image features that serve as components of dynamically generated object models [11], and the use of this structure for matching novel components. These features are represented by patches of information derived from object views of the ETH-80 image set [12]. During learning, the relationships between a given set of image features extracted from training object views are automatically established through an unsupervised learning process according to their similarity. Throughout recall, these relationships are exploited by the classifier in order to retrieve the best matching model features for a given set of image features extracted from a test object view.

The Neural Map's learning procedure uses the GNG algorithm to approximate the topology of the feature distribution. However, this algorithm is highly parameterized and the selection of the parameter values considerably affects its performance. Currently, they are determined empirically based on the parameter values proposed in [13] together with the ones resulting from preliminary experiments, which apply the GNG algorithm to a subset of the ETH-80 image set. This selection mechanism is time consuming and requires prior knowledge about the feature distribution. Furthermore, its resulting values potentially lead to suboptimal solutions. Consequently, a more robust selection mechanism is required in the Neural Map's learning procedure.

Evolutionary Algorithms (EAs) are a class of stochastic search algorithms grounded on the Darwinian principles of evolution [14], [15]. In this optimization paradigm, the candidate solutions are represented as individuals of a population based on GNG networks trained with a subset of the ETH-80 image set. The fitness of these individuals for enduring into future generations is measured using three different approaches: *Global Error*, *Sample Distance*, and *Sample Distance with Growth Restriction*. The population is

Guillermo S. Donatti is with the International Graduate School of Neuroscience, the Ruhr-University Research School, and the Institut für Neuroinformatik, Ruhr-Universität, 44780 Bochum, Germany (email: guillermo.sebastian.donatti@neuroinformatik.rub.de).

Oliver Lomp is with the Institut für Neuroinformatik, Ruhr-Universität, 44780 Bochum, Germany (email: oliver.lomp@neuroinformatik.rub.de).

Rolf P. Würtz is with the International Graduate School of Neuroscience, the Ruhr-University Research School, the Department of Electrical Engineering and Information Sciences, and the Institut für Neuroinformatik, Ruhr-Universität, 44780 Bochum, Germany (email: rolf.wuertz@neuroinformatik.rub.de).

Guillermo S. Donatti and Oliver Lomp contributed equally to this work.

978-1-4244-8126-2/10/$26.00 

evolved with mechanisms inspired by biological evolution (i.e., reproduction, mutation, and recombination) and using two different starting conditions: *random* and *empirical*. The different sets of parameter values found with this selection mechanism are cross-validated using the object categorization and recognition experimental protocol introduced in [9].

The detailed description of the methods used for the present work is organized as follows: Section II defines how EAs are utilized for optimizing the parameter values of the GNG algorithm; Section III details the mechanism used to find the fittest individuals for the different approaches proposed in the evolutionary optimization scheme; Section IV describes the experiments performed to cross-validate the performance of the different sets of parameter values obtained during optimization; and Section V discusses the experimental results and outlines future research steps.

## II. Optimizing parameter values

Following the principles of natural selection, EAs find local optima of a fitness function by evolving a set of individuals over the course of many generations. For this purpose, the concept of natural selection is formalized into the evolutionary cycle. Before entering the cycle, the first parent population is initialized and evaluated, assigning each individual a fitness value based on its performance. At the beginning of the cycle, the current parent population mates to create the offspring population, which is then mutated and evaluated. Afterwards, the fittest individuals in the current parent and offspring populations are selected to form the new parent population for the next generation. This process repeats until a termination criterion is met (e.g., a preset number of generations, or a given time constraint).

### A. Individuals

In the present work, an individual is composed of a set of candidate values for the parameters of the GNG algorithm, which are detailed in Table I. During evolutionary optimization over $g$ generations with a population of $\rho$ members, a generic individual $I_{i,j} \in \mathbb{R}^{n_r} \times \mathbb{Z}^{n_z}$ with $0 \leq j < \rho$, $0 \leq i < g$ and $n_r, n_z \in \mathbb{N}$, is defined by

$$I_{i,j} = (\mathbf{r}_{i,j}, \mathbf{z}_{i,j}) \ , \tag{1}$$

where $\mathbf{r}_{i,j} \in \mathbb{R}^{n_r}$ and $\mathbf{z}_{i,j} \in \mathbb{Z}^{n_z}$ are vectors of real and integer values, respectively, which represent the parameters of the GNG algorithm. These values, also referred to as entries, delineate the genome of the individual $I_{i,j}$. The order of the parameter values in the individual's genome is arbitrary and follows the one in Table I. In order to simplify the notation, the indices $i$ and $j$ are dropped and only $I$, $\mathbf{r}$ and $\mathbf{z}$ are used to denote individuals and their entries.

### B. Initialization

We use two methods for initializing the individuals in the parent population. The first one assigns a random value, drawn uniformly from the interval permitted for the respective parameters, to each entry in the individuals. The second method assigns a predetermined set of values to all individuals in the initial parent population. These values are empirically determined starting with the ones used for a completely different task in [13] and further modifications motivated by preliminary experiments on a subset of the ETH-80 image set.

| Parameter | Domain | Description |
|---|---|---|
| $\epsilon_I$ | $\mathbb{R}$ | winner neuron learning rate |
| $\epsilon_n$ | $\mathbb{R}$ | neighbor of winner learning rate |
| $\alpha_g$ | $\mathbb{R}$ | error decay on growth |
| $\alpha_d$ | $\mathbb{R}$ | error decay rate |
| $\kappa$ | $\mathbb{R}$ | threshold for neuron deletion |
| $a_{\max}$ | $\mathbb{Z}$ | maximal age of synapse |
| $s_{\max}$ | $\mathbb{Z}$ | maximal number of synapses |
| $n_{\max}$ | $\mathbb{Z}$ | maximal number of neurons |
| $\lambda_{\text{growth}}$ | $\mathbb{Z}$ | growth time interval |
| $\lambda_{\text{decay}}$ | $\mathbb{Z}$ | decay time interval |

TABLE I

THE GNG ALGORITHM PARAMETERS COMPRISED BY THE INDIVIDUALS. THE UPPER BOUNDS FOR NUMBER OF NEURONS $n_{\max}$ AND SYNAPSES $s_{\max}$ ARE INTRODUCED TO AVOID INTRACTABLE NETWORKS.

Empirically determined parameter values may be used during initialization to condition the search space and avoid locally optimal parameter value sets, which result in GNG networks without the capability of adapting to the topology of the input distribution. The presence of many such parameter value sets as well as the resulting behavior is suggested by preliminary experiments conducted in [16].

### C. Mating procedure

The mating of two parent individuals, which gives rise to a new individual, is accomplished through *mating selection* and *recombination*. During the former, the parents are randomly selected from the current parent population with a uniformly distributed probability. The latter is achieved by applying the one-point-crossover operator [14], [17] to the real and integer values. This operator starts copying values from the first parent individual and switches to copying from the second one at a point randomly selected with a uniform distribution. Consequently, the genome of the resulting offspring individual is composed of entries from the genomes of both parents.

### D. Mutation scheme

While mating is used to combine parameter values from the parents, mutation serves the purpose of exploring previously unused ones.

Integer values are mutated using the discrete mutation function $m_D : \mathbb{Z}^{n_z} \mapsto \mathbb{Z}^{n_z}$ proposed in [14], wherein each entry in the vector $\mathbf{z}$ has a preset probability of undergoing mutation. If mutation occurs, a random value within the parameter's range is selected and set as its new value.

The mutation of real-valued parameters is based on the scheme described in [14], which adds a random number

drawn from a normal distribution with mean zero and variance $\sigma^2$ to each entry contained in the vector $\mathbf{r}$. However, instead of using a single variance, individual variances are associated with each entry in the genomes to reflect the different sizes of the permitted ranges for the parameter values. This method allows for a finer exploration of smaller parameter spaces (e.g., parameters that have values in the range $[0, 1]$) and a broader mutation for parameters with larger spaces. The values for the individual variances are based on empirical observations made in [16].

Having an individual $I$, let the vector of associated variances for the mutation of real-valued parameters be $\mathbf{s} = \left(\sigma_1^2, \ldots, \sigma_{n_r}^2\right)$ and $\mathbf{x} = (x_1, \ldots, x_{n_r})$ be a vector where each element $x_k$, with $0 \leq k < n_r$, is a normally distributed zero-mean random variable $x_k \sim \mathcal{N}\left(0, \sigma_k^2\right)$. Then, the mutation function $m : \mathbb{R}^{n_r} \times \mathbb{Z}^{n_z} \mapsto \mathbb{R}^{n_r} \times \mathbb{Z}^{n_z}$ of an individual $I$ is given by

$$m\left(I\right) = \left(\mathbf{r} + \mathbf{x}, m_D(\mathbf{z})\right) . \tag{2}$$

### *E. Evaluating and selecting individuals*

Ruled by the principles of natural selection, the evolutionary optimization process considers only the $\mu$ fittest individuals to populate a new generation. The elitist selection scheme [17], a combination of the $(\mu, \lambda)$ and $(\mu + \lambda)$ selection schemes described in [18], is used to select these individuals from the parent and the offspring population, based on a fitness value assigned to each of them during a fitness assessment.

In order to evaluate the fitness of an individual $I$, the GNG algorithm is applied to a training subset of the ETH-80 image set using the individual's genome to configure the parameters of its learning process. Let $U_I = \{\mathbf{n}, \mathbf{s}\}$ be the undirected graph of the resulting GNG network, where the neurons are represented as the node vector $\mathbf{n} = (n_1, \ldots, n_N)$ and the synapses as the edge vector $\mathbf{s}$, as well as the neuron centers $\mathbf{c}_l$ in the input space with $1 \leq l \leq N$. Upon completion of the training, the resulting GNG network is used by a fitness function $f : \mathbb{R}^{n_r} \times \mathbb{Z}^{n_z} \mapsto \mathbb{R}$ to assign a fitness value to the individual. This fitness value indicates how well the resulting GNG network topology approximates the topology of a subset of the ETH-80 image set.

The present work uses three fitness functions, each of which takes into consideration a different approach for evaluating the performance of the GNG algorithm's learning process. The proposed fitness functions are calculated using error measures. Therefore, the EA considers an individual fitter than another if its assigned fitness value is lower.

*1) The Global Error fitness function:* This fitness function uses the accumulated error variable tracked for each neuron by the GNG algorithm to calculate the fitness of an individual. With the nodes $\mathbf{n}$ of the GNG network $U_I$, the accumulated error $e_l$ (calculated according to [5]) of $n_l$, the Global Error fitness function is calculated as the sum of all accumulated errors:

$$f_{GE}(I) = \sum_{l=1}^{N} e_l \, . \tag{3}$$

The parameters of the GNG algorithm that directly influence the accumulation of the neurons' error values (i.e., $\alpha_g$ and $\alpha_d$) are not optimized when using the Global Error fitness function because preliminary experiments performed in [16] suggest that they always evolve to a solution that minimizes the network's global error independent of how well its topology approximates the one of the sample data distribution.

*2) The Sample Distance fitness function:* The approach proposed for the fitness function $f_{SD}$ is not directly influenced by any parameters of the GNG algorithm. Instead, a testing subset $S_T$ of the ETH-80 image set is used to evaluate the fitness of an individual $I$. This approach allows the evolutionary optimization process to overcome the limitations present when using the Global Error fitness function. Consequently, all available parameters can be optimized when using the Sample Distance fitness function.

Given the trained GNG network $U_I$ and the test set $S_T$ of cardinality $T$, the fitness value assigned by the Sample Distance fitness function is calculated by

$$f_{SD}\left(I\right) = \sum_{\mathbf{t} \in S_T} d\left(\mathbf{c}_{\tilde{n}}, \mathbf{t}\right) , \tag{4}$$

where $\tilde{n}$ is the closest neuron to $\mathbf{t}$ in $U_I$, according to the distance measure $d$ calculated using the normalized scalar product as shown in Equation 5, with $\mathbf{x}$ and $\mathbf{y}$ being two vectors of the same dimension.

$$d\left(\mathbf{x}, \mathbf{y}\right) \; = 1 - \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} \, . \tag{5}$$

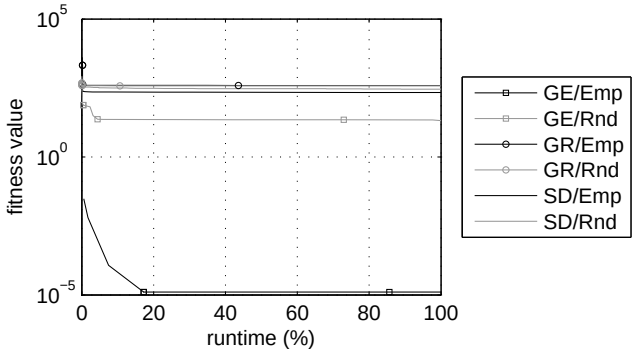
*3) The Growth Restriction for the Sample Distance fitness function:* The fitness of an individual measured with the Sample Distance method described in Section II-E.2 favors the selection of individuals with genomes that lead to GNG networks with large amount of neurons, since having more neurons reduces the size of the Voronoi regions of all neurons and hence also the sum of their distances to the test samples [5]. During the evolutionary optimization process, using this fitness function eventually leads to individuals that generate GNG networks with one or more neurons for each available sample, thus having over-fitted the problem or being intractable in terms of computational complexity.

Grounded in these observations, a growth restriction based on the total number of neurons is imposed upon the Sample Distance fitness function:

$$f_{GR}(I) = f_{SD}(I) + N \, . \tag{6}$$

## III. Selecting the fittest individuals for cross-validation

The core of the evolutionary optimization scheme introduced in Section II is the nature of the individuals' search space, together with the evaluation and selection of the best parameter value set encoded by the individuals' genomes. As described in Section II-E, the prior can be either *empirical* (Emp) or *random* (Rnd) while three fitness functions are used for the latter: *Global Error* (GE), *Sample Distance* (SD), and *Sample Distance with Growth Restriction* (GR).




(a)

| Approach | Total Generations | Last Imp. Generation |
|---|---|---|
| GE/Emp | 182 | 174 |
| GE/Rnd | 44333 | 7521 |
| SD/Emp | 582 | 502 |
| SD/Rnd | 134227 | 88407 |
| GR/Emp | 38788 | 16888 |
| GR/Rnd | 90490 | 87488 |

(b)

Fig. 1. The summary of the evolutionary optimization process using the different combinations of starting conditions and fitness functions introduced in Section II. (a) shows the progression of the fitness values of the fittest individuals. The time axis is scaled as percent of the runtime. (b) shows the total number of generations for each evolutionary process and the number of the generation in which the last improvement occurred. The implementation of these experiments is based on the *SHARK* library [17].

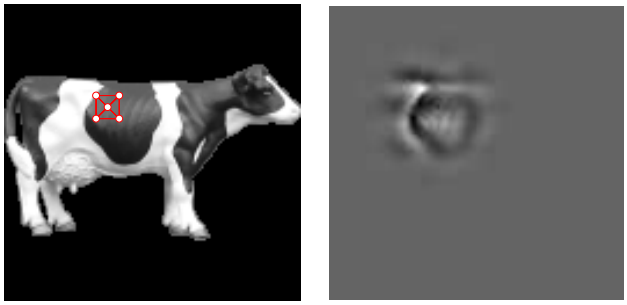
The fittest individual for each combination of these different approaches is selected through an evolutionary optimization process. During this process, the generations are composed of four individuals, two from the parent population and two from the offspring one. The purpose of finding such representatives lies in the use of their genomes to cross-validate the evolutionary optimization scheme (see Section IV).

As it would occur in nature for different animal species, the amount of generations varies according to the time each individual requires to be developed and evaluated under each evolutionary approach. Consequently, the stopping condition of the optimization process is only time based. The results of these experiments are summarized in Figure 1, and each of their main components is detailed further in the remainder of this section.

### A. The data sets

During the individuals' evaluation process, the fitness function assigns a value to an individual by assessing a GNG network trained with its parameter values to represent the visual object knowledge captured by a subset of object views from the ETH-80 image set. This image set contains views and segmentation masks of 80 objects within a taxonomy composed of 8 basic level categories (i.e., cows, dogs, horses (Hor.), apples, pears, tomatoes (Tom.), cars, and cups) from 4 superordinate areas (i.e., animals (Anim.), fruits and vegetables (F.&V.), human made big (H.m.b.), and human made small (H.m.s.)). Each category contains 10 different individuals that are represented by 41 images from viewpoints spaced equally over the upper viewing hemisphere. In the present work, these views and their respective segmentation masks are combined to generate segmented gray value images.

Fig. 2. Samples of ETH-80 object views extracted at 90° vertically and all the horizontal angles taken at intervals of 22.5°. They are segmented gray value images of single objects scaled to 128×128 pixels. The first row depicts samples from the eight view-points used for training; the second row shows the corresponding ones used for testing.



(a) (b)

Fig. 3. *Square* image features: (a) depicts a scheme of the local feature detector placed over an ETH-80 object view; (b) illustrates the reconstruction [20] of the derived *Square* image feature.

The training and test data sets used during this process are composed of *Square* image features [9] derived from a subset of the ETH-80 image set. This subset contains the object views extracted from the 16 view-points defined by the vertical angle of 90° and all the horizontal angles taken at intervals of 22.5°. The features obtained from half of the view-points are used for training and the remaining ones are used for testing. Figure 2 shows an example of this partitioning.

The *Square* image features are obtained using a local feature detector [9], which consists of the complex responses from five Gabor jets [19] derived with a square-shaped structure from ETH-80 images. It represents a medium-sized patch of information extracted from object views. Figure 3 exemplifies this type of image feature.

### B. Training and testing procedures

The evaluation process of the individuals comprises a training and a test procedure (see Section II-E). In the first one, the GNG algorithm uses the training data set to incrementally develop a neural network. In the second one, the test data set is used to measure the performance of the trained GNG network.

On the one hand, the evaluation process under all the evolutionary optimization approaches uses the same training procedure. During training, the algorithm's learning process is tailored to the parameter values encoded by the individual's genome. This procedure is considered complete after 110 epochs, where each sample from the training data set is used once per epoch by the algorithm's learning process. The topology of the resulting GNG network approximates the one of the training *Square* image features.

On the other hand, the test procedure is tied to the nature of the fitness function used by each of the evolutionary optimization approaches. In the case of the GE fitness function, the GNG network's performance is measured with the sum of the accumulated errors registered by its neurons during training. The test set is neglected in this case. Alternatively, the SD fitness function employs the trained GNG network for identifying the neuron, which is closest to each of the test *Square* image features according to the normalized scalar product of their values. In this alternative case, the sum of the smallest distance measure values is used to test the efficacy of the trained GNG network. As stated in Section II-E, the test procedure when using the Growth Restriction is achieved similarly to the one of the SD fitness function by adding to the resulting error measure the total neuron number of the trained GNG network.

Figure 4 depicts the components of the GNG network's mean error (i.e., the global error and the total number of neurons) observed during the training procedure of the fittest individuals from the evolutionary optimization approaches.

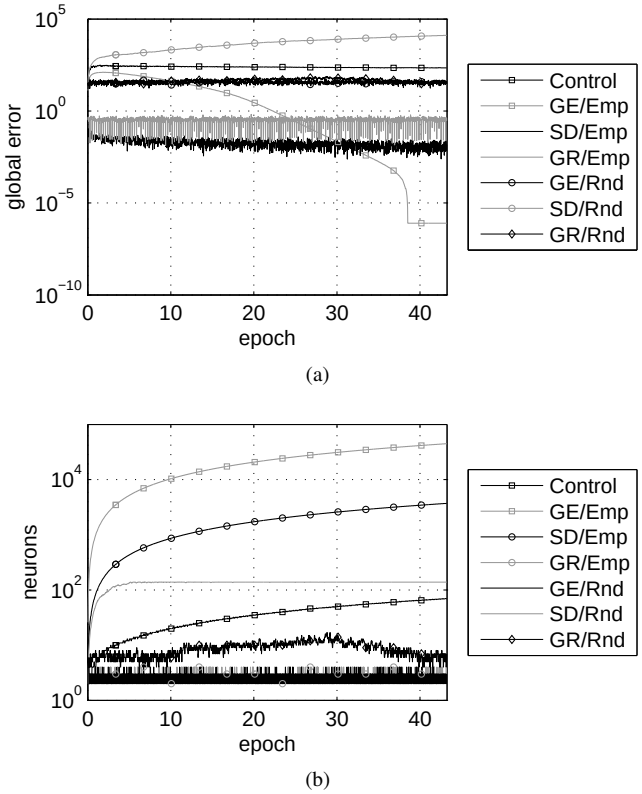



Fig. 4. The mean errors' components of the GNG networks observed during the epochs of the training procedure: (a) depicts the sum of the neurons accumulated errors and (b) illustrates the total number of neurons. These neural networks are obtained with the parameter values encoded by genomes of the fittest individuals.

## IV. Cross-validating the evolutionary optimization scheme

The feasibility of using the scheme proposed in Section II to improve the parameter values for the GNG algorithm (comprised in the Neural Map's learning procedure) is evaluated with the experiments introduced in [9]: *view-point invariance test* and *leave-one-out cross-validation*. This analysis encompasses the cross-comparison and cross-validation of this scheme's fitness functions.

### A. Invariant object recognition and categorization tests

The experiments propounded in [9] test the performance of a feature-based object recognition and categorization system with object views of the ETH-80 image set. This system is composed of a Neural Map and a winner-take-all voting scheme. During a learning procedure, image features from training object views are used to shape a self-organized structure of model features [9]. After this training stage completes, image features extracted from a test object view are matched against the Neural Map's model features. Finally, the registered responses are subjected to a voting scheme that consolidates the recognized object superordinate and basic-level categories. The object identity is also recognized during this recall procedure, but only in the view-point invariance test.

### B. Novel feature retrieval tests

The experimental protocol detailed in [9] is also replicated without the voting scheme. The purpose of these additional experiments is to measure the Neural Map's ability to use the information acquired throughout training for novel feature retrieval. In these experiments, the Neural Map's learning procedure is achieved similarly to the ones described in Section IV-A. However, during the recall procedure, the matching responses are considered successful when the novel and the model feature labels are equal, without subjecting these responses to a winner-take-all voting scheme.

### C. Experimental results

The experiments outlined in Section IV-A and Section IV-B employ the parameter values encoded by the genomes of the fittest individuals (see Section III) and control ones given by [9] to train and test seven Neural Maps.

As stated in Section II-E.3, the parameter values found with the Sample Distance fitness functions and empirical starting conditions lead to GNG networks that are intractable in terms of computational complexity. This is also observed experimentally when using the Global Error fitness functions with the same starting condition. Therefore, in these cases the neural growth is limited to the maximal amount of neurons

| Approach | Superordinate Level | | Basic Level | | Object Identity | | Neurons | Synapses | | Error | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Voting | Yes | No | Yes | No | Yes | No | Total | Max. | Min. | Global | Mean |
| Control | 89.88 | 70.77 | 77.38 | 51.92 | 48.81 | 17.40 | 1506 | 40 | 2 | 155.69 | 0.10 |
| GE/Emp (L1) | 88.86 | 70.43 | 76.11 | 51.40 | 48.02 | 17.09 | 792 | 37.20 | 2 | 208.88 | 0.26 |
| GE/Emp (L2) | 89.69 | 70.83 | 76.65 | 52.02 | 49.71 | 17.52 | 1584 | 36.60 | 2 | 192.21 | 0.12 |
| GE/Rnd | 84.39 | 67.65 | 71.54 | 48.41 | 41.29 | 14.87 | 177.60 | 16.80 | 1.80 | 191.09 | 1.08 |
| GR/Emp | 59.62 | 44.37 | 34.34 | 22.29 | 7.48 | 3.13 | 3 | 2 | 1 | 0.02 | 0.01 |
| GR/Rnd | 71.28 | 58.22 | 58.88 | 36.74 | 29.23 | 10.13 | 11 | 5.80 | 1.60 | 63.33 | 5.76 |
| SD/Emp (L1) | 92.02 | 71.21 | 77.43 | 52.09 | 48.50 | 17.54 | 792 | 25.60 | 1 | 0.04 | 0 |
| SD/Emp (L2) | **92.42** | **71.68** | **77.60** | **52.71** | **50.25** | **18.03** | 1584 | 30 | 1 | 0.03 | 0 |
| SD/Rnd | 88.40 | 69.73 | 74.38 | 50.58 | 47.85 | 17.16 | 261.20 | 15 | 2 | 2347.81 | 8.99 |

TABLE II

AVERAGED RESULTS OF EXPERIMENTS BASED ON THE VIEW-POINT INVARIANCE TEST WITH (SECTION IV-A) AND WITHOUT (SECTION IV-B) VOTING. THE FIRST SIX COLUMNS SPECIFY THE PERCENTAGE OF CORRECT CATEGORIZATION/RECOGNITION. FURTHERMORE, THE NUMBER OF NEURONS, THE MAXIMAL AND MINIMAL NUMBER OF SYNAPSES PER NEURON, AND THE GLOBAL AND MEAN ERRORS ARE SPECIFIED.

observed in the control case (L2), and to half of this size (L1).

All results are obtained by averaging several trials with the experimental set-up introduced in [9] under different starting conditions. The robustness of the experiments starting conditions is achieved by shuffling the training data sets during the Neural Map's learning procedure.

*1) View-point invariance test:* The training and test data set partitioning for the experiments grounded on the view-point invariance test is based on the object's view-points. In this case, the test object views are selected by rotating horizontally training object views by $22.5°$. Likewise to the approach used in the evolutionary optimization process (see Section III-A), $1680$ object views from $21$ view-points are used during learning and $1600$ from $20$ view-points are used during recall.

The learning procedure of the Neural Maps is considered complete after $1000$ epochs. In one epoch, the Neural Maps' underlying GNG neural network is adapted using $79236$ *Square* image features derived from the training object views. During recall, each test object view is used once, and a total of $74930$ *Square* image features are matched during this procedure.

The properties of the trained the Neural Maps are averaged and rounded for each of the seven evaluated approaches. The structural and performance information derived from their underlying GNG networks together with their categorization and recognition rates are summarized in Table II.

*2) Leave-one-out cross-validation:* The experiments based on the leave-one-out cross-validation use a data set partitioning based on the object's identity. In these experiments, the Neural Maps are trained using the image features extracted from all $41$ object views of $79$ objects, and the ones derived from the $41$ object views of the remaining object are used for testing.

The learning procedure of the Neural Maps in these experiments finishes after $125$ epochs. The amount of training and test *Square* image features used throughout the learning and recall procedures depend on the data set partitioning. Overall, the Neural Maps' underlying GNG neural network is adapted during one epoch using between $150919$ and $153184$ image features. The test object views are employed once during recall, matching amid $982$ and $3247$ image features during this procedure. These procedures are repeated for all $80$ partitions of the ETH-80 image set, and the rates are averaged over all tests.

The resulting categorization rates registered by these experiments are shown in Table III. These results are cross-validated with previous results on the same image set using alternative methods. These methods use texture information extracted from the object views to categorize a novel object. On the one hand, the $D_x D_y$ and the Mag-Lap are variations of histograms of local gray value derivatives at multiple scales [12]; the prior is a rotation-invariant descriptor and uses first derivatives in a two dimensional coordinate system; the latter uses the gradient magnitude and the Laplacian. On the other hand, the combination of a feature-based and a correspondence-based approaches (FB-MBC) in a form of graph dynamics based on the hypothesis that the brain's data structure has the form of dynamic graphs whose nodes are labeled with elementary features [21]; in this case the method is evaluated using hierarchically organized (H) and single element (S) categories.

## V. CONCLUSIONS AND FUTURE RESEARCH

The present work employs the optimization paradigm detailed in Section II to identify the parameter values of the GNG algorithm comprised in the Neural Map's learning procedure [9]. This paradigm is based on EAs and explores six different optimization approaches given by the combination of three fitness functions and two starting conditions (see Section III). The parameter values obtained from the fittest

| Approach | Superordinate Level | | Basic Level | |
|---|---|---|---|---|
| Voting | Yes | No | Yes | No |
| Control | **97.50** | 64.73 | **88.75** | 41.24 |
| GE/Emp (L1) | 96.25 | 63.09 | 78.75 | 39.50 |
| GE/Emp (L2) | **97.50** | 64.69 | 83.75 | 41.08 |
| GE/Rnd | 85 | 58.82 | 72.50 | 35.67 |
| GR/Emp | 56.25 | 43.10 | 26.25 | 19.87 |
| GR/Rnd | 83.25 | 54.59 | 67 | 31.05 |
| SD/Emp (L1) | 96.25 | 64.46 | 83.75 | 40.95 |
| SD/Emp (L2) | **97.50** | 65.47 | 86.25 | 41.86 |
| SD/Rnd | **97.50** | 64.95 | 87.50 | 41.37 |
| $D_xD_y$ | N/A | | 79.79 | |
| Mag-Lap | N/A | | 82.23 | |
| FB-MBC (H) | 49.18 | | 64.07 | |
| FB-MBC (S) | N/A | | 74.57 | |

| Approach | Anim. | H.m.b. | H.m.s. | F.&V. | Cows | Dogs | Hor. | Cars | Cups | Apples | Pears | Tom. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Control | **96.67** | **100** | **90** | **100** | 80 | 60 | 70 | **100** | **100** | **100** | **100** | **100** |
| GE/Emp (L1) | **96.67** | **100** | 80 | **100** | 60 | 30 | 40 | **100** | **100** | **100** | **100** | **100** |
| GE/Emp (L2) | **96.67** | **100** | **90** | **100** | 90 | 40 | 40 | **100** | **100** | **100** | **100** | **100** |
| GE/Rnd | 83.33 | 70 | 60 | **100** | 50 | 30 | 40 | 80 | **100** | **100** | 90 | 90 |
| GR/Emp | 33.33 | 0 | 50 | **100** | 0 | 0 | 0 | 10 | 70 | 80 | 0 | 50 |
| GR/Rnd | 73.33 | **100** | 70 | **100** | 30 | 20 | 10 | **100** | 90 | **100** | 80 | **100** |
| SD/Emp (L1) | **96.67** | **100** | 80 | **100** | 80 | 40 | 60 | **100** | **100** | **100** | 90 | **100** |
| SD/Emp (L2) | **96.67** | **100** | **90** | **100** | 80 | 50 | 60 | **100** | **100** | **100** | **100** | **100** |
| SD/Rnd | **96.67** | **100** | **90** | **100** | 90 | 60 | 50 | **100** | **100** | **100** | **100** | **100** |
| $D_xD_y$ | N/A | N/A | N/A | N/A | 82.68 | 62.44 | 58.78 | 98.29 | 66.10 | 85.37 | 90 | 94.63 |
| Mag-Lap | N/A | N/A | N/A | N/A | **94.39** | **74.39** | **70.98** | 77.56 | 77.80 | 80.24 | 85.37 | 97.07 |
| FB-MBC (H) | 80 | 6.10 | 4.88 | 47.48 | 48.10 | 37.64 | 56.71 | 47.46 | 54.83 | 90.25 | 89.95 | 87.64 |
| FB-MBC (S) | N/A | N/A | N/A | N/A | 49.76 | 35.85 | 57.80 | 80.98 | 98.05 | 91.22 | 87.80 | 95.12 |

TABLE III

OBJECT CATEGORIZATION PERCENTAGES FOR LEAVE-ONE-OUT CROSS-VALIDATION WITH (SECTION IV-A) AND WITHOUT (SECTION IV-B) VOTING. THE RESULTS OF THE $D_xD_y$ AND THE MAG-LAP RECOGNITION METHODS DESCRIBED IN [12], AND THE ONES OF THE FB-MBC APPROACH INTRODUCED IN [21], ARE INCLUDED FOR COMPARISON.

individuals of these approaches are cross-compared and cross-validated with empirical ones using the experimental protocol introduced in [9].

The results detailed in Section IV indicate that the SD/Emp approach maximizes the Neural Map's performance in the experiments based on the view-point invariance test. This approach outperforms all the other ones, even when having a neuronal growth limit set to approximately half the size observed for a control GNG network (see Section IV-C.1). Nevertheless, this result is not replicated in the experiments grounded in the leave-one-out cross-validation. Despite the facts that in these experiments both SD/Emp and SD/Rnd achieve the highest novel feature retrieval rates, even when the parameter values are not optimized for this task, this improvement is not sufficient to change the voting scheme verdicts. Therefore, their categorization rates are slightly inferior only to the control ones. In general, the remaining approaches performed below the control ones in all experiments, with the exception of the Mag-Lap that performed slightly better only in the animal basic-level categories. In this particular case, the false positives are concentrated in the same superordinate area for the control, SD, GE, and GR/Rnd approaches. These results suggest that the empirically set parameter values can be further optimized using the given optimization paradigm with the SD/Emp approach. However, the improvement of the Neural Map's performance using the

optimized parameter values may depend on the difference between the conditions used for the data set partitioning in the optimization paradigm's evaluation process, and the ones from the Neural Map's experiments.

It is worth noting that the evolutionary optimization process with random starting conditions favors individuals with small values for $n_{\max}$ and $s_{\max}$. Furthermore, it favors individuals with zero valued $\epsilon_I$, when using $f_{GE}$; and likewise for the $\epsilon_n$ parameter, when utilizing $f_{SD}$ and $f_{GR}$. The approaches that use the $f_{GE}$ select individuals with very small $\lambda_{\text{growth}}$ values, which lead to rapidly growing GNG networks.

The search for the fittest individuals (see Section III) encounters two limitations that should be resolved in future research. The first one relates to the computational complexity of training and evaluating large GNG networks using high dimensional image features. This is observed particularly when employing a combination of fitness functions and starting conditions like the GE/Emp or the SD/Emp, which favor individuals that lead to large GNG networks. The evolution of a generation using these processing-intensive approaches can become computationally intractable. The second limitation is the poor ability of the fitness functions to favor the saliency of individuals that may generate better approximations of the feature distribution topology. The present work introduces the Growth Restriction (see Section II) to overcome the first limitation, but empirically it is too restrictive, and Neural Maps using this approach are experimentally the least performing ones. As an alternative, the number of generations needed to find an optimal individual can be reduced with a method based on the Covariance Matrix Adaptation (CMA) [22], [23]. However, CMA only optimizes continuous values and, therefore, has to be adapted for discontinuous ones in order to be used with the approach presented in this paper. Employing CMA might also allow for the use of the CMA for Multi-objective Optimization [24] to restrict the growth rates of the GNG networks. Concerning the second limitation, the optimization process may be improved using additional fitness functions to assess the learning capabilities of a GNG network (e.g., assigning lower fitness values for errors generated in later generations and disregarding errors in earlier ones, or incorporating a rate of change observed in the global error curve of the GNG algorithm).

## ACKNOWLEDGMENTS

We gratefully acknowledge funding from the European Commission in the *NovoBrain* project (MEST-CT-2005-020385); from the land of Northrhine-Westphalia in the project *Mobile Vision System* (w0806it041), which is co-financed by the ERDF from the European Commission, from the DFG (WU 314/5-3); and the Ruhr-University Research School funded by Germany's Excellence Initiative (DFG GSC 98/1).

## REFERENCES

[1] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59–69, 1982.

[2] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Prentice Hall, July 1998.

[3] T. Martinetz and K. Schulten, "Topology representing networks," *Neural Networks*, vol. 7, no. 3, pp. 507–522, 1994.

[4] B. Fritzke, "A growing neural gas network learns topologies," in *Advances in NIPS*, G. Tesauro, D. S. Touretzky, and T. K. Leen, Eds. MIT Press, 1995, pp. 625–632.

[5] ——, "A self-organizing network that can follow non-stationary distributions," in *ICANN*, 1997, pp. 613–618.

[6] M. Pagel, E. Maël, and C. von der Malsburg, "Self calibration of the fixation movement of a stereo camera head," *Machine Learning*, vol. 31, pp. 169–186, 1998.

[7] P. Li, I. Farkas, and B. MacWhinney, "Early lexical development in a self-organizing neural network," *Neural Networks*, vol. 17, no. 8-9, pp. 1345–1362, 2004.

[8] H. S. Loos and C. von der Malsburg, "1-click learning of object models for recognition," in *Biologically Motivated Computer Vision*, H. H. Bülthoff, S.-W. Lee, T. A. Poggio, and C. Wallraven, Eds. Springer, 2002, pp. 377–386.

[9] G. S. Donatti and R. P. Würtz, "Using Growing Neural Gas networks to represent visual object knowledge," in *21st IEEE International Conference on Tools with Artificial Intelligence*. IEEE Computer Society, 2009, pp. 54–59.

[10] T. Trappenberg, *Fundamentals of Computational Neuroscience*. Oxford University Press, June 2002.

[11] G. Westphal and R. P. Würtz, "Combining feature- and correspondence-based methods for visual object recognition," *Neural Computation*, vol. 21, no. 7, pp. 1952–1989, 2009.

[12] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," in *Computer Vision and Pattern Recognition*, 2003, pp. II–409–415.

[13] B. Bolder, *Sensomotorische Koordination eines Roboterkopfes*. Shaker, Aachen, 2006.

[14] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[15] T. Bäck, "Evolutionary algorithms in theory and practice," Ph.D. dissertation, University of Dortmund, Department of Computer Science, February 1994.

[16] O. Lomp, "Finding optimal parameters for neural networks using evolutionary algorithms," B.Sc. Thesis, ET-IT Dept., Univ. of Bochum, Germany, Oct 2008.

[17] C. Igel, V. Heidrich-Meisner, and T. Glasmachers, "Shark," *The Journal of Machine Learning Research*, vol. 9, pp. 993–996, 2008.

[18] H. Beyer and H. Schwefel, "Evolution strategies–A comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.

[19] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," *IEEE Transactions on Computers*, vol. 42, no. 3, pp. 300–311, 1993.

[20] M. Günther and R. P. Würtz, "Reconstruction of images from graphs labeled with Gabor jets," 2010, in preparation.

[21] G. Westphal, "Feature-driven emergence of model graphs for object recognition and categorization," Ph.D. dissertation, University of Lübeck, Germany, 2006.

[22] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.

[23] T. Suttorp, N. Hansen, and C. Igel, "Efficient covariance matrix update for variable metric evolution strategies," *Machine Learning*, vol. 75, no. 2, pp. 167–197, 2009.

[24] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evolutionary Computation*, vol. 15, no. 1, pp. 1–28, 2007.