

Sub-goal based Robot Visual Navigation through Sensorial Space Tesselation

George Palamas

Faculty of Computing, Engineering and Science
University of South Wales
Wales, United Kingdom

J. Andrew Ware

Faculty of Computing, Engineering and Science
University of South Wales
Wales, United Kingdom

Abstract—In this paper, we propose an evolutionary cognitive architecture to enable a mobile robot to cope with the task of visual navigation. Initially a graph based world representation is used to build a map, prior to navigation, through an appearance based scheme using only features associated with color information. During the next step, a genetic algorithm evolves a navigation controller that the robot uses for visual servoing, driving through a set of nodes on the topological map. Experiments in simulation show that an evolved robot, adapted to both exteroceptive and proprioceptive data, is able to successfully drive through a list of sub-goals minimizing the problem of local minima in which evolutionary process can sometimes get trapped. We also show that this approach is more expressive for defining a simplistic fitness formula yet descriptive enough for targeting specific goals.

Keywords—Self Organization; Growing Neural Gas; Genetic Algorithm; Robot Map Building; Visual Servoing; Recurrent Neural Network

I. INTRODUCTION

With respect of vision based robot navigation, most research work is focused on four major areas: map building and interpretation; self-localization; path planning; and obstacle-avoidance. Of these four major research areas, self-localization is of key importance. The recognition of the initial position, the target position, and the current position occupied by the robot while wandering around are all bound to a self-localization process. The main two approaches used for robot localization are landmark based and appearance based techniques. In this paper, we describe a combination of a developmental method for autonomous map building and an evolutionary strategy to verify the results of the map interpretation in terms of navigation usability.

Our strategy involves two discrete phases: map building and navigation phase. In the first phase an agent freely explores a pre-determined simulated terrain, collecting visual signatures corresponding to positions in the environment. After the exploration, a self-organizing algorithm builds a graph representation of the environment with nodes corresponding to known places and edges to known pathways.

During the second phase, a population of robot controllers is evolved to evaluate map usability. Robots evolve to autonomously navigate from an initial position to a goal position. In order to facilitate successful translation, a shortest path algorithm is employed to extract the best path for the robot to follow. This algorithm also reveals all those intermediate

positions that the robot needs to traverse in order to reach the goal position. These intermediate positions act also as sub-goals for the evolution process.

II. SENSING THE ENVIRONMENT

To be fully autonomous, a robot must rely on its own perceptions to localize. Perception of the world generates representation concepts, topological or geometrical, within a mental framework relating new concepts to pre-existing ones [3]. The space of possible perceptions available to the robot for carrying out this task may be divided into two categories: Internal perception (proprioception) or perceptions of its own interactions with the world, associate changes of primitive actuator behavior like motor states; external or sensory perception (exteroception) is sensing things of the outside world. A robot's exteroceptors include all kinds of sensors such as proximity detectors and video cameras. Our system uses only visual information for map building and navigation.

III. ROBOT NAVIGATION

Landmark-based localization methods rely on the assumption that landmarks can be detected and accurately interpreted from raw sensor readings [2], [5]. However interpretation from sensor readings to accurate geometric representation is complex and error prone. On the contrary, an appearance-based representation of an environment is not encoded as a set of geometrical visual features, but as an appearance map that includes a collection of sensor readings obtained at known positions [1]. The advantage of this representation is that the raw sensor readings generate a qualitative estimate of position. The currently perceived image can be directly matched with past experiences stored in the appearance-based topological representation [6]. This method of using sensor readings does not rely on precise metric measurements as with traditional geometrical based maps.

In the field of computer vision the use of appearance based techniques has become widespread. A comparison between the two families of vision based localization methods can be found in [4], showing that appearance-based methods are more robust to noise, occlusions and changes in illumination, when compared to landmark based-methods. The source of inspiration for such techniques comes from the animal kingdom. Small animals, such as insects, navigate through natural environments seemingly with little effort. For example, despite their relatively simple nervous system (and hence limited memory capacity), bees and desert ants are able to

retrace their movements. Such a level of efficiency indicates flexible representations of the surroundings based on visual cues taken from target locations such as home and food sources [7]. These representations seem to have an appearance based flavor rather than a Cartesian arrangement of landmarks. To visit target locations after prior exploration, insects traverse in a way that reduce discrepancies between the stored snapshot and their current retinal image [8].

As stated already the main drawback of appearance-based methods is that localization is only possible in previously mapped areas. Several successful applications have shown promising results [9],[10]. Like landmark based mechanisms, appearance based navigation systems suffer from the problem of perceptual aliasing [11], the situation that different locations produce identical sensory perceptions. A possible solution could be the incorporation of temporal or odometry information to resolve any conflicts. Another possible solution is to divide the goal into a set of sub goals of smaller tasks easier to fulfill. Such an approach, even if perceptual aliasing is present, is more efficient since subtasks are easier to manage and achieve.

IV. ENVIRONMENT REPRESENTATION

The most natural representation of a robot's environment is a map. In addition to representing places in an environment, a map may include other information, such as properties of objects, regions that are unsafe or difficult to traverse, together with information of prior experience. An internal representation of space can be used by a robot to pre-plan and pre-execute tasks that may be performed later.

A. Geometric Representation

Geometric maps are quantitative representations made up of discrete geometric primitives such as lines, polynomial functions, points and so forth. They are characterized by large scale detail. The primary shortcoming of geometrical model based representation relates to the fact that they can be difficult to infer reliably from sensor data [12].

B. Topological Representation

A topological map is one which captures the connectivity of the environment and has been simplified so that only vital information remains and unnecessary detail has been removed. These maps lack geometric information such as scale, distance and orientation but the relationship between points is maintained. The simplicity of topological maps support much more efficient planning than metric maps [13],[29].

The key to a topological relationship is based on an abstraction of the environment in terms of connectivity between discrete regions or objects, with edges connecting them. In the simplest form, this may involve a complete absence of metric data. A robot employing this representation has no real understanding of the geometric relationship between locations in the environment but the enclosed information is sufficient for the robot to conduct point to point motion. The use of graphs has been exploited in many robotic systems to represent spaces. The following example [10] is representative.

A graph is a kind of abstract data structure that consists of points or nodes connected by links, called lines or edges. Each node corresponds to one of the unique landmarks and each edge corresponds to known paths between them. If the environment consists of networks of corridors and rooms (as found in many indoor environments, such as office buildings or hospitals), it is less complex to specify the topology of important locations and their connection suffice.

Humans represent physical spaces topologically rather than geometrically. For example, when providing the clues needed to lead someone in a building, directions are usually of the form "go down the hall, turn right at the elevator, open the second door on your left," rather than in geometric form. Topological maps are sparse representations of the environment as a collection of visual feature vectors at certain positions. Such representations present some advantages difficult to ignore. First and foremost the computational and memory cost is relatively low. The path planning in metric maps can be computationally very expensive; unlike the lightweight planning nature of graph based structures. Second, they do not require accurate determination of robot's position and therefore are less sensitive to error accumulation, commonly occurring in metric mapping approaches. Topological visual navigation is usually based on key-frame matching to self-localize and navigate to a previously visited location [14, 15].

V. MAP-BUILDING PHASE

A. Terrain Exploration

Our approach considers robots to be like insects, equipped with simple control mechanisms tuned to their environments. Therefore, a model of terrain exploration using a simple two dimensional Brownian random walk was implemented. Such an approach could mimic the navigation behavior of simple animals and microorganisms such as insects. Random walk (also known as Brownian motion) is a process that consists of a sequence of steps, in which the direction and size of each move is determined randomly. The advantage of this approach is minimization of simulation artifacts such as cyclic behavior. During this step the robot collects panoramic snapshots at regular time intervals.

B. Visual Feature Extraction

This collection of panoramic images represents a large amount of raw data and therefore it is necessary to extract some specific features that describe the content of each image. Color histograms are a very appealing graphical representation. Image analysis based on color information is robust for robot map-building and image retrieval problems and, due to their statistical nature, provide a complete rotationally invariant representation when employed with panoramic cameras (figure 1).

Moreover, they are also computationally cheap to implement. Omni-directional vision systems are a special type of vision sensor. Images are obtained by placing a convex mirror a short distance from a camera. These systems provide a 360° view of the robot's environment around the vertical axis.

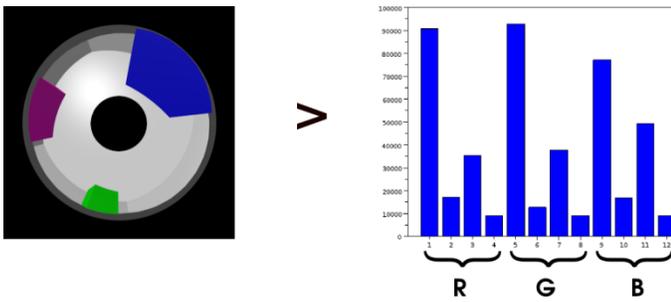


Fig. 1. Omnidirectional snapshot and extracted RGB histograms.

The set of color signatures, extracted during terrain exploration, can be manipulated as a large abstract image database. This is the foundational scheme of a content based image retrieval approach. Self localization can be based on a measure of resemblance between the currently acquired image of the robot and the base of images stored as perceptual signatures representing familiar terrain. Gonzalez and Lacroix [18] suggests a qualitative position refinement technique that localize a rover when it comes back to a previously perceived area, using an image indexing technique on panoramic views based on principal component analysis. The limitation of this procedure is that it cannot perform incrementally, because all learning images are required to compute the subspace.

To measure color histogram similarity, we use the standard Euclidean formula. This distance metric is a comparison between the identical bins in the respective histograms and all bins contribute equally to the distance [19]. The Euclidean distance between two color histograms h and g is given by

$$d_E(h, g) = \sum_{m=0}^{M-1} (h[m] - g[m])^2 \quad (1)$$

C. Self-Organization of Visual Signatures

In unsupervised learning networks the only data available is the input set. These networks serve two main purposes: topology preservation and vector quantization. Topology preservation means that close input signals are mapped to neurons which are close in the network and conversely, close neurons in the network come from close input signals in the input space, preserving similarities between data as much as possible (figure 2).

There are many reasons why we use a self organizing system for robot mapping, preferred over other mechanisms that have no plasticity properties [20]. The first reason is that less parameters, which describes the robot operation, need to be predetermined. Information given by sensors incorporate noise, leading to erroneous conclusions regarding spatial perception. Information may be contradictory when sensor readings come from different sensors but represents the same robot position. Furthermore, due to the nature of sensors used with respect to the task being performed, extracted information may not be useful. Self-organizing mechanisms make use of all available data without prior assumptions. Data clustering addresses the problem of noise and handles meaningless information.

One of the most robust algorithms is the Growing Neural Gas (GNG) by Fritzke [21]. Growing Neural Gas is a network that can learn the topological relationships from an input set of vectors using a variation of the Hebbian rule. GNG dynamically add or remove nodes and can approximate the input space with higher accuracy compared to a network with predefined structure (figure 2) such as the Kohonen self organizing feature map [22]. The GNG is an adaptive algorithm inspired by the physical properties of uniform gases and the work on self organizing maps. Assuming that a given distribution of points is represented by a container shape, the algorithm will begin to create freely moving particles which will try to expand uniformly to fill the input space. After convergence is reached, the network nodes then represent the shape of the container.

Clustering of static sensor signals has been used before for robot localisation [13],[23]. Different unsupervised neural network architectures have been used to realize topological relationships between input and output space. Baldassari et.al. [24], applied a GNG algorithm for a visual based self-localisation task of a mobile robot in an indoor environment. Images acquired from a camera moving in a pathway, formed an implicit topological representation of the environment. These simulations dictated the effectiveness of the GNG model in recognition speed, classification tasks and in particular topology representation as compared to the popular Kohonen Self Organizing Map (SOM) model. This performance gap, ascribed to the fact that GNG algorithm, that dynamically add and remove nodes, can approximate the input space more accurately than a network with a predefined structure and size - such as a SOM. This is true also since SOM resembles a lossy compression scheme by applying a data projection from a multidimensional space, where perceptual signatures are described, to preferably only a two dimensional space.

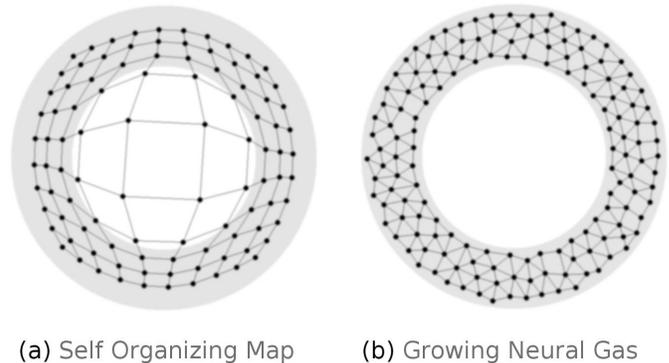


Fig. 2. Topology preservation for SOM and GNG respectively.

VI. THE NAVIGATION PHASE

A. Path Planning

Prior to navigation, path planning is an important issue as it directs the robot on how to get from an initial position to a goal position. Since the environment is stationary with no other moving obstacles, the process of path planning is straightforward.

Topologically, this problem is equivalent to the shortest path problem of finding a route between two nodes in the graph. Many algorithms have been developed to find a path in a graph. For example, Dijkstra's algorithm [25] computes the optimal path between a single source point to any other point in a graph (figure 3). Since we compute the path once after the mapping phase, a real time algorithm is not necessary.

B. Self-localisation

The robot continuously keeps track of the current location. While the robot moves, collects snapshots and exports corresponding color histograms. Every newly acquired histogram is being compared with every stored histogram in the graph structure. The robot self-localizes when the closest histogram on the topological map is found. Each of the nodes in the graph represents a specific histogram and the closest one indicates the current position of the robot on the map.

C. Visual Navigation

For the robot to conduct point to point navigation, a controller is necessary that will move the robot through a set of intermediate points towards the final position. The proposed robot behavior controller realizes an Elman neural network (Elman NN) and a genetic algorithm (GA). Neural network architectures are particularly well suited for complex pattern classification tasks and genetic algorithms are good optimization procedures because they can explore large and multidimensional spaces to find global solutions. Hence, they are well suited for training neural networks.

The neural controller is composed of a grid of input neurons whose activations are given by the color bins of the corresponding histograms. Two output neurons control the angular torque applied to the left and right wheel of the robot.

A set of neurons with recurrent connections fed from hidden and output neuron layer, help to learn past instances and correlate them with new information. The input neurons of the neural network are activated by sensory data, and the output neurons control the motors of the robot. Within a population, each individual has a different genome describing a different neural network (different weight vectors), thus resulting in specific individual responses to sensory-motor interactions with the environment. These behavioral differences affect the robot's fitness, which is defined, by the number of successive milestones traversed by the robot.

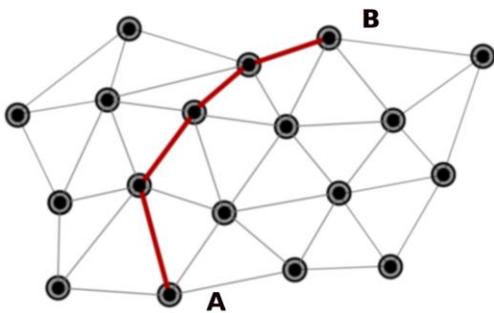


Fig. 3. Dijkstra's graph search algorithm output.

Evolutionary strategies require that a large population of individuals be evaluated over the course of many generations. In the case of evolutionary robotics it has been assumed that it would take far too long to do all of these evaluations in the real world. The main practice is to evaluate in simulation, whether partial or in whole. The aim of this evolutionary strategy is to create a population of agents with different genomes, each defining a set of parameters of the control system of the robot. The genome is this set of parameters whose translation into a phenotype, the actual behavior of the controller, can cause the system to depict biological behaviors. The artificial genome decodes the weight values associated to synaptic connections of an artificial neural network that determines the global visual navigation behavior.

D. Neural Network Controller

The neural network we use (figure 4) is a typical feed-forward architecture with evolvable thresholds and discrete-time, fully-recurrent connections at the output layer [26]. This type of neural network is used to do sequence processing, especially when these sequences are made of indexed data [28]. The processing occurs in steps and it is assumed that neuron outputs are computed instantaneously. A set of twelve input neurons receive information about the color distribution from the images captured from the panoramic camera.

Each neuron covers a band of the color variations in the image that is a bin value is assigned to each input. Each of the RGB color components of the image are divided into four bands. The activation of each neuron is scaled in the interval [0, 1] so that activation 0.5 corresponds to zero torque applied in the wheels. Activation values above and below 0.5 stands for forward and backward rotational speeds, respectively. The two output neurons act also as proprioceptive information about the speed of each wheel. A set of short term memory units stores the values of the output neurons at the previous sensor-motor state and sends them back to the output units through a set of recurrent connections [26]. All other neurons in the hidden layer have recurrent connections to store previous activity.

$$f(x) = 1/(1 + \exp(-x)) \quad (2)$$

Neurons use the sigmoid activation function in the range [0,1], where x is the weighted sum of all inputs (equation 2). For each discrete time interval they encode both the sensorial information and the motor commands passed to the wheels.

E. Evolving Controllers

Algorithms in Evolutionary Robotics (ER) frequently operate on populations of candidate controllers, initially selected from some random initial population of controllers. This population is then repeatedly modified according to a fitness function, a particular type of objective function that is used to indicate the closeness of a given design solution to achieving a set of aims.

Evolutionary Robotics builds upon several aspects of artificial evolution. The Genetics aspect is about what goes into the artificial chromosomes and how these chromosomes are mapped into individuals. Genetic encoding and genotype-phenotype mappings are the key to the evolvability of a system.

In our case the genotype represents the architecture of a controller in a form of a binary string and the phenotype represents the possible solution space. The population of robot controllers is also referred to as genomes.

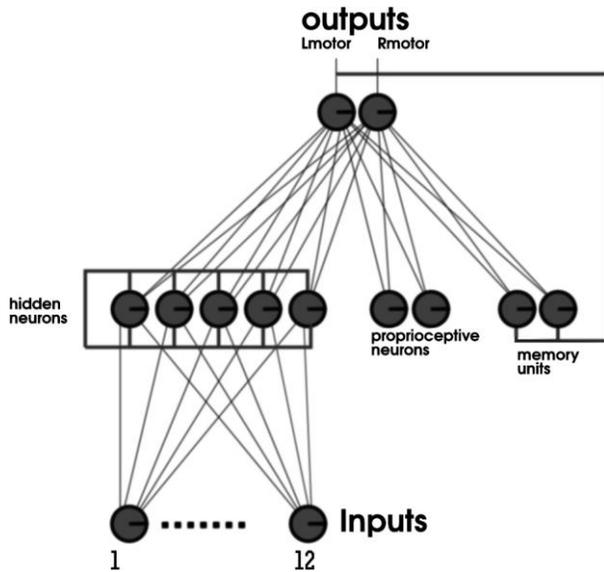


Fig. 4. Discrete-time recurrent neural network

Evolutionary algorithms have been widely used to design cognitive architectures for robots with emergent behaviors (see [16, 17] for an overview). The main strength is their ability to cope well with high complexity problems using only a high-level reward function. Best candidates are rewarded only for their global efficiency because of the impossibility of foreseeing every sub-goal the robot has to solve. If the global objective is very hard then initial performance may be so poor that the evolutionary process is hard to initiate. Another problem is local minima in which the evolutionary process may become trapped. A fitness function must be simplistic yet descriptive enough for targeting specific goals. Designing a fitness function is essential to the successful use of a genetic algorithm. If the fitness function is poorly designed, the algorithm will either converge on an inappropriate solution, or will have difficulties in converging at all.

For a successful incremental evolution process the system requires an accurate knowledge of the problem to be solved so as to lead the evolutionary algorithm to perfect convergence. For graph based robot navigation the global task can be divided into smaller tasks. Both global task and sub-tasks are self-similar, i.e. the goal is to transfer the robot from one point to another. Since in our case the different sub-tasks are in nature exact copies of the main task, by just dividing the path that the robot needs to traverse, the only requirement is to determine when to switch from one sub-task to another. Fitness function is an objective function used as a metric to calculate the distance of each individual from a set of goals.

The success of evolutionary algorithms depends on the fitness function design.

A good function design must guarantee that a collection of solutions exists, differentiating enough, with values that changes neither too rapidly nor too slowly with the given parameters of the optimization problem. The fitness function was designed to select robots for their ability to arrive at the goal zone. The neural network has a set of evolvable connections that are individually encoded in the genome. A population of 100 individuals is randomly initialized and each individual genome is decoded into the synapses of the neural network. The twenty percent of the population with the highest values are used for reproduction and the rest discarded. The new genomes have a crossover value of 0.1 per pair and mutation probability of 0.01. The meaning of crossover is swapping a pair of genetic strings around a randomly chosen point. Mutation consists of toggling the value of a random bit in the genetic sequence. The best two genomes from the previous generation are inserted to the current generation, unaltered, to improve the stability of the process. This strategy is known as elitist selection.

F. The Evolution Process

The fitness function was designed to select the best robots to arrive at the goal node and is described as follows. The fitness value is the percentage of the distance the robot covered between two adjacent nodes in the path. Every time the robot reaches a node in the node sequence, as extracted from the path planning phase, it is rewarded with a value of 1. Since it is extremely difficult for the robot to match the current perceived histogram with the target node, we made the assumption that 90% of the covered distance corresponds to successful goal reaching.

The robot must traverse the nodes in the specific order as dictated from the outcome of the Dijkstra's algorithm. If the robot arrives at a goal node that is not successive in order, the robot is not awarded for this sub-goal. Successful individuals have to arrive at all sub-goal nodes through this specific order. The running fitness value for every agent in the population is the summation of extra value gained for each successive step plus the current percentage of the distance between currently arrived at node and the next one in the sequence.

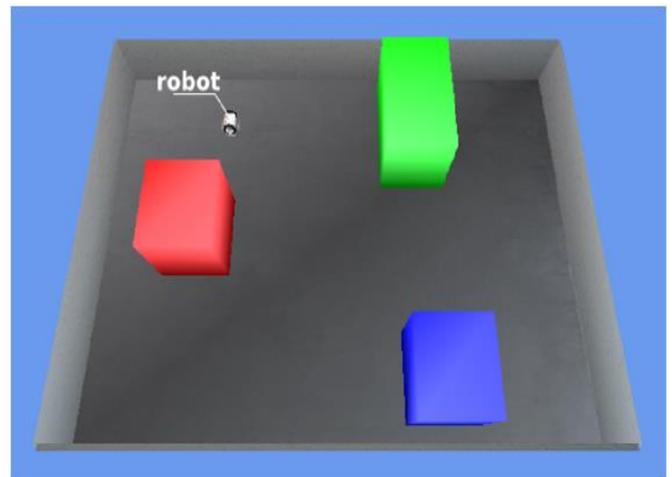


Fig. 5. The simulated environment and the robot used in our experiments

VII. EXPERIMENTAL SETUP

A. The Robot

The simulated robot can be seen in figures 5. The omnidirectional camera is widely used in visual based robot navigation and localization, which is due to the large field of view. Images are obtained by placing a convex mirror a short distance from a camera. The main advantage that led us to promote this solution is the large field of view compared to orthographic or standard cameras. The system provides a 360° view of the robot's environment around the vertical axis when it is mounted on top of the robot. Landmarks are always in the field of view except for occasional occlusion and therefore have increased reliability. This is advantageous when utilizing topological representations as the more information the image contains the more stable it is. Another advantage is the orientation independency when employed with statistical methods such as color histograms.

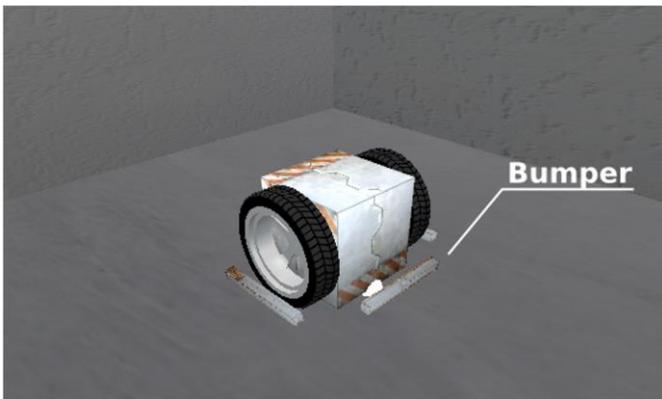


Fig. 6. Simulated differential drive robot. The robot has four bumpers to detect collisions with walls and obstacles.

The robot is cubical in shape with two independent drive wheels attached in the middle of the chassis and two trailing casters, front and rear. This is a typical differential drive setup and the robot can change its direction by varying only the relative rotation speed of its wheels and hence does not require an additional steering mechanism. The robot is equipped with two, one bit, horizontal axis bumper bars. The purpose of the tactile sensors, when a reaction to a collision occurs, is to reposition the next individual to initial conditions and start a new simulation trial.

B. The Environment

For the experiments we used a simple 3D world, a closed rectangular arena with colored obstacles, dark gray walls and no ceiling (figure 6). This environment is not as visually complex as a typical real life environment. The primary goal was to demonstrate the plausibility of evolving agents that could use cognitive maps and behaviors based on visual information which otherwise would be very difficult if not impossible to employ.

Both for the robot and the simulated environment we used the Bullet physics libraries, a freely available software package that models gravity, mass, friction, and collisions. For modeling the environment as well as the view from the omnidirectional camera, we used the XNA graphics

framework. Well understood image processing tasks such as color histogram extraction were accomplished using the well known open Computer Vision (openCV) library. To model the reflective surface of the sphere we applied a method of environmental mapping known as Cube Mapping [27]. This is a technique for approximating the appearance of a reflective surface by means of a pre-computed texture image. The image is generated, for every simulation step, by projecting the surroundings of the sphere onto the six faces of a cube. This cubical texture is then wrapped onto the sphere to represent reflection lighting properties.

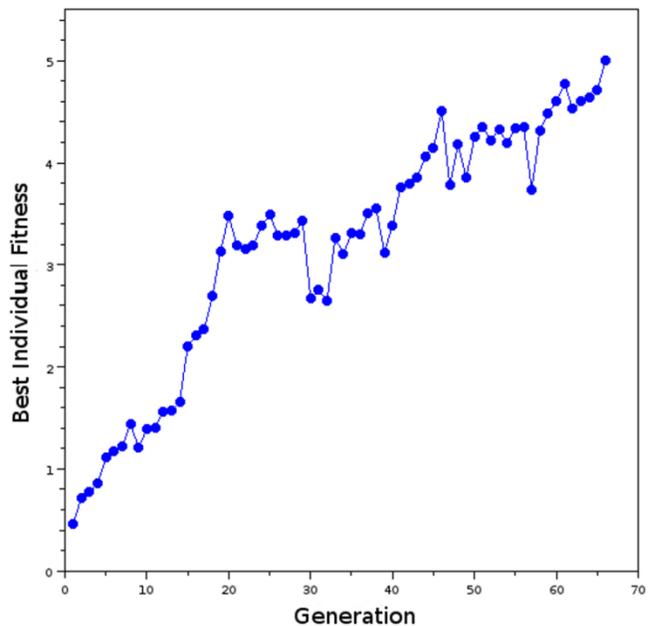


Fig. 7. Best individual fitness value from each generation.

C. Experimental Procedure

Initially a robot is allowed to freely navigate in the environment in order to build a collection of 500 panoramic snapshots of the environment from different perspective views. A GNG algorithm, performing off-line, formed a grid with topological relations between these visual cues. The grid starts with only two nodes and grows until the criterion of 20 nodes is met. Based on this grid, a shortest path is extracted to indicate optimal route from a starting position to a global target position in the arena. A genetic algorithm evolved a neuro-controller to allow a robot to successively follow the six nodes the optimal path consists of. Each individual robot tested for a period of time lasting for 10 seconds or 1000 simulation cycles. Trials were truncated earlier if collisions detected from the bumpers.

VIII. RESULTS

This section shows experimental results of the proposed method. Several sets of experiments were performed with varying parameters relating with the GNG algorithm, the neural network architecture and the genetic algorithm. Something worth mentioning is the fact that the dark shades of the environment gave better results than other colorations. This is simple to explain since black color interprets as absence of color and does not interfere with the three other landmarks, the

discrete nature of which is being enhanced. Figure 7 depicts a record of the best individual score for each generation to evaluate the solution domain. As can be seen, a navigation controller evolved after 66 generations. The robot that used this controller, managed to pass through all the intermediate points until the final objective. The path followed by the robot is shown in Figure 8. The gray and red points correspond to intermediate sub-goals and final goal respectively. The optimal path planning computed with Dijkstra's algorithm between an initial and final position in the graph that was generated by the GNG algorithm.

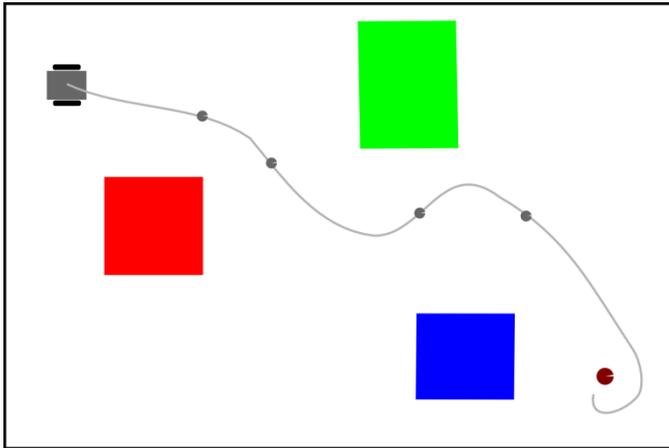


Fig. 8. The robot successfully followed the sequence of nodes. The small gray dots are the positions that the robot encountered the threshold of 90% of the distance covered between two adjacent nodes. (A successful controller is always awarded with a fitness value of 5). The difference between the actual position and the robot position is due to the error in the calculation of the best matching unit and the 90% accuracy threshold.

IX. CONCLUSIONS

This paper explores the advantages of evolutionary sub-goal robot navigation with a cognitive map architecture. All methods used have been tested using a simulated environment. The GNG algorithm has been previously shown to be effective in forming topological maps through an appearance based framework. Evolutionary strategies have also been applied successfully in solving complex problems such as visual navigation. However these algorithms may take some time to converge to an optimal solution. Feature selection is a particularly important step for building robust learning models. Our method is based on global only image properties and may suffer from the problem of perceptual aliasing [30], the fact that different physical locations correspond to similar sensory perceptions.

However the purpose of this study was to demonstrate the efficiency of simple algorithms to solve complex systems. After verification of the aforementioned algorithms using simulations, these need to be evaluated on actual robots and modify as necessary to ensure acceptable real life robot navigation.

Further research might explore alternative visual scene interpretation methods for dealing with more complex navigation scenarios.

ACKNOWLEDGMENT

The authors would like to thank all laboratory members for their valuable discussions through this research.

REFERENCES

- [1] I. Ulrich and I. Nourbakhsh, Appearance-based place recognition for topological localization, vol. 2, no. April. Ieee, 2000, pp. 1023–1029.
- [2] S. Se, D. G. Lowe, and J. J. Little, Vision-based global localization and mapping for mobile robots, vol. 21, no. 3. IEEE, 2005, pp. 364–375.
- [3] A. Asciani, E. Frontoni, A. Mancini, and P. Zingaretti, “Feature group matching for appearance-based localization,” in IEEEERSJ International Conference on Intelligent Robots and Systems, 2008.
- [4] R. Sim and G. Dudek, “Comparing image-based localization methods,” in Pattern Recognition, 2003, vol. 18, pp. 1560–1562.
- [5] J. J. Leonard and H. F. Durrant-Whyte, Mobile robot localization by tracking geometric beacons, vol. 7, no. 3. IEEE, 1991, pp. 376–382.
- [6] B. J. A. Kröse, N. Vlassis, and R. Bunschoten, “Omnidirectional Vision for Appearance-based Robot Localization,” in Int Workshop on Sensor Based Intelligent Robots, 2002.
- [7] T. S. Collett and S. P. D. Judd, “Multiple stored views and landmark guidance in ants,” Nature, vol. 392, no. 6677, pp. 710–714, 1998.
- [8] B. A. Cartwright and T. S. Collett, “How honey bees use landmarks to guide their return to a food source,” Nature, vol. 295, no. 5850, pp. 560–564, 1982.
- [9] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.
- [10] M. O. Franz, B. Schölkopf, H. A. Mallot, and H. H. Bülthoff, “Learning view graphs for robot navigation,” Autonomous Robots, vol. 5, no. 1, pp. 111–125, 1998.
- [11] C. Siagian and L. Itti, Biologically Inspired Mobile Robot Vision Localization, vol. 25, no. 4. IEEE, 2009, pp. 861–873.
- [12] H. H. Gonzalez-Banos and J. C. Latombe, “Navigation Strategies for Exploring Indoor Environments,” The International Journal of Robotics Research, vol. 21, no. 10–11, pp. 829–848, 2002.
- [13] M. V. Butz, E. Shirinov, and K. L. Reif, “Self-Organizing Sensorimotor Maps Plus Internal Motivations Yield Animal-Like Behavior,” Adaptive Behavior, vol. 18, no. 3–4, pp. 315–337, 2010.
- [14] O. Booij, B. Terwijn, Z. Zivkovic, and B. Krose, “Navigation using an appearance based topological map,” Proceedings 2007 IEEE International Conference on Robotics and Automation, no. April, pp. 3927–3932, 2007.
- [15] Y. W. Y. Wang, D. C. D. Chen, and C. S. C. Shi, Autonomous Navigation Based on a Novel Topological Map, vol. 1. Ieee, 2009, pp. 254–257.
- [16] S. Nolfi, “Evolving robots able to self-localize in the environment: the importance of viewing cognition as the result of processes occurring at different time-scales,” Connection Science, vol. 14, no. 3, pp. 231–244, 2002.
- [17] I. Harvey, E. D. Paolo, R. Wood, M. Quinn, and E. Tuci, “Evolutionary Robotics: A New Scientific Tool for Studying Cognition,” Artificial life, vol. 11, no. 1–2, pp. 79–98, 2005.
- [18] J. J. Gonzalez-Barbosa and S. Lacroix, Rover localization in natural environments by indexing panoramic images, vol. 2, no. May. Ieee, 2002, pp. 1365–1370.
- [19] J. R. Smith and S. Chang, “Tools and techniques for color image retrieval,” Image Rochester NY, vol. 2670, no. 3, pp. 426–437, 1996.
- [20] U. Nehmzow, “Map building through self-organisation for robot navigation,” in Advances in Robot Learning 8th European Workshop on Learning Robots EWL8 Proceedings Lecture Notes in Artificial Intelligence Vol1812 SpringerVerlag Berlin Germany, 2000, vol. 1812, pp. 1–22.
- [21] B. Fritzke, “A Growing Neural Gas Network Learns Topologies,” Advances in Neural Information Processing Systems 7, vol. 7, no. 1, pp. 625–632, 1995.
- [22] T. Kohonen, “Self-organized formation of topologically correct feature maps,” Biological Cybernetics, vol. 43, no. 1, pp. 59–69, 1982.

- [23] S. Yamada, "Recognizing environments from action sequences using self-organizing maps," *Applied Soft Computing*, vol. 4, no. 1, pp. 35–47, 2004.
- [24] P. Baldassarri, P. Puliti, A. Montesanto, and G. Tascini, "Self-Organizing Maps versus Growing Neural Gas in a Robotic Application," in *Proceedings of the 7th International Work-Conference on Artificial and Natural Neural Networks: Part II: Artificial Neural Nets Problem Solving Methods*, 2003, pp. 201–208.
- [25] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [26] D. Floreano, M. Suzuki, and C. Mattiussi, "Active Vision and Receptive Field Development in Evolutionary Robots," *Evolutionary Computation*, vol. 13, no. 4, pp. 527–544, 2005.
- [27] N. Greene, *Environment Mapping and Other Applications of World Projections*, vol. 6, no. 11. IEEE, 1986, pp. 21–29.
- [28] J. L. Elman, "Finding Structure in Time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.
- [29] S. Thrun, "Robotic Mapping: A Survey," *Science*, vol. 298, no. February, pp. 1–35, 2002.
- [30] A. Angeli, S. Doncieux, J. A. Meyer, and D. Filliat, *Incremental vision-based topological SLAM*. Ieee, 2008, pp. 22–26.