Proceedings of the 2001 IEEE Congress on Evolutionary Computation Seoul, Korea • May 27-30, 2001

Evolutionary Calibration of Sensors using Genetic Programming on Evolvable Hardware

Ho-Sik Seok
Artificial Intelligence Lab (SCAI)
School of Computer Science and Engineering
Seoul National University
Seoul 151-742, Korea
hsseok@scai.snu.ac.kr

Byoung-Tak Zhang
Artificial Intelligence Lab (SCAI)
School of Computer Science and Engineering
Seoul National University
Seoul 151-742, Korea
btzhang@scai.snu.ac.kr

Abstract- In order to retain some degree of decision-making ability in a complex and dynamic environment, there are many attempts to build autonomous mobile robots. However, conventional methods pay little attention to the unreliability of sensors. Because of the corruption by noise and the difference of sensitivity, even the same kind of sensors shows different observation under the same condition. This causes a problem that a minor change of the environment of the sensor system has great influence on the perception ability of the robot. To improve the reliability of sensors, we present a method for evolutionary calibrating sensors using genetic programming as calibration mechanism. In our approach, sensor calibration logic is implemented on evolvable hardware. Therefore, as the learning goes on, sensor interpretation circuit reconfigures itself to a more suitable form during runtime. Through two experiments on different tasks, we confirmed that our method improved the correctness of interpretation significantly.

1 Introduction

Most robots employ specialized controllers that are carefully designed by hand, using knowledge of the robot, its environment and the task. This approach faces a couple of serious problems. Some of the prior knowledge like certain aspects of the robot dynamics or the characteristics of the robot's sensors is usually hard to obtain, and making domain knowledge computer-accessible often requires tremendous amounts of programming time[21]. Because of these difficulties, the mobile robot must retain some degree of decision-making ability in a complex and dynamic environment. Several attempts have been proposed to build autonomous mobile robots[4, 16, 17, 20, 21]. To improve robot performance, many researchers tried to optimize the sensor system of robots[1, 12, 13, 14]. These approaches can be classified as sensor evolution. Sensor evolution is a kind of phylogenetic learning process by which basic categories of perception are selected and refined[5]. However, these approaches have one problem. They pay little attention on the lack of reliability of typical sensors. The existing sensors often are not capable of directly measuring the quantity of interest. Furthermore, sensor measurements are typically corrupted by noise. Often, the distribution of this noise is not known[20]. We notice the lack of reliability. To improve the reliability of sensors, we propose a method to calibrate sensors using evolutionary method. In the proposed method, genetic programming is used to calibrate sensors. Due to the tree structure of genetic trees, it is easy to incorporate prior knowledge. So genetic programming can improve search procedure by using the result of other experiment. In our approach, sensor calibration logic is implemented on evolvable hardware. Because of its dynamic reconfigurability, evolvable hardware can offer considerably higher performance while adapting to a changing environment[18]. By adoption of evolvable hardware, we implement a specified logic for sensor calibration that reconfigures its logic during runtime. We apply our method to two kinds of experiments. In experiment 1, we evolve interpretation logic of sensors according to the sensitivity of each sensor under the same condition. In experiment 2, we evolve interpretation logic of one sensor under the different conditions. Using our method, we succeed at reducing error rate considerably in each experiment. This paper is organized as follows. Section 2 explains automatic sensor calibration using genetic programming. Section 3 illustrates our method for sensor evolution and implementation detail.In Section 4, some experimental results are shown. Section 5 reviews related work. Section 6 discusses future work.

2 Motivation of Automatic Sensor Calibration

Autonomous mobile robots perceive their environments through sensors. But even the same kind of sensor reports very different data. Table 1 is a good example of such problem. These data were sensed using sensors of a Khepera robot[23]. Khepera robot perceives its environment through eight IR sensors. Each sensor reports its observed data in the range of from 0 to 1023. Figures 2 and 3 illustrate the measurement environment. Table 1 is the data measurement by the sensors for environment 1. Table 2 is the observed data of environment 2.

As known from the graphs and tables, sensors of a Khepera robot are very unreliable. Even at the same condition, the observed data have very large deviation. This causes the following problems. First, each sensor is likely to output different estimation under the same condition. Second, due to

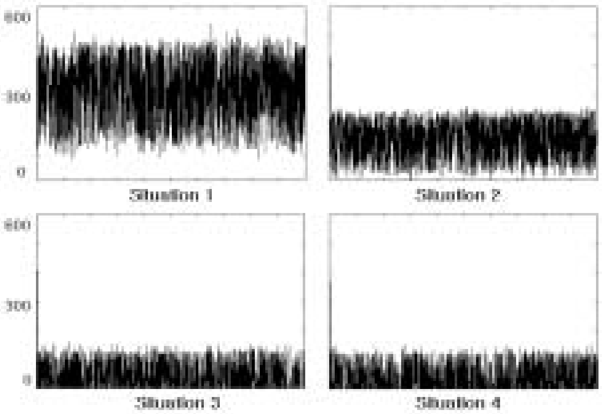

Figure 1: Sensed data of environment 1

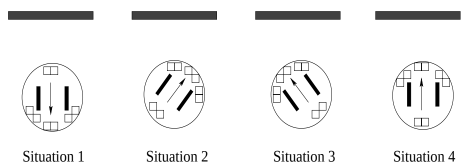

Figure 2: Experimental environment 1-Under the same condition, only the observing sensor is changed

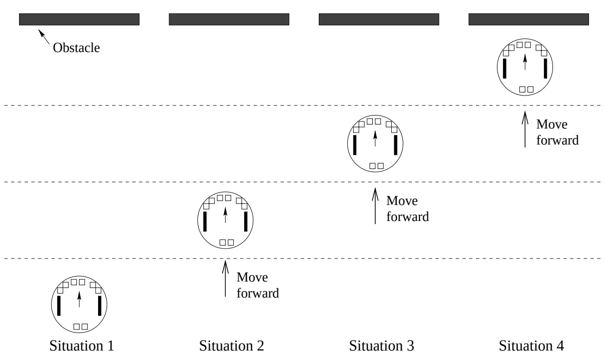

Figure 3: Experimental environment 2-Using the same sensor, only the environment is changed

Table 1: Experimental environment 1
Ave=Average, SD=Standard Deviation

Situation	1	2	3	4
Ave	236.02	187.62	59.91	71.14
SD	146.67	117.88	77	79

Table 2: Experimental environment 2

Situation	1	2	3	4
Ave	316.5	140.35	43.47	39.21
SD	75.59	76.81	53.29	51.31

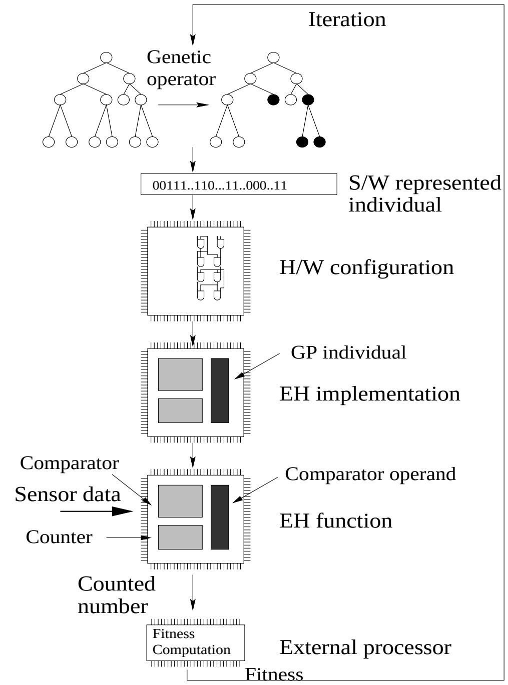

Figure 4: Sensor evolution procedure

the overlapped section, it is difficult to determine robot's correct location. Third, even a minor change of environment or sensor configuration has great influence on the perception ability of a robot. To address these problems, it is necessary to calibrate each sensor automatically.

3 Evolving Genetic Programs for Sensor Calibration

Figure 4 illustrates the detailed sensor calibration procedure. Sensor calibration is performed on evolvable hardware (EH) and an external processor.

The tree structure is converted into a binary string representation for implementation on evolvable hardware. The hardware-represented individual compares sensor data with the values from the max and min sub-trees. The number of correctly measured data is counted. The external processor maintains information on individual configuration and performs arithmetic calculations like division. Configuration information of GP individuals on evolvable hardware is implemented in hardware form by a control program on the external processor. Figure 5 shows an example GP tree. In order to determine the interpretation section of sensed data, genetic trees w ith two subtrees named as the max tree and the min tree maintain the maximum and minimum values of the section. If the environment or sensor system changes, the max tree and the min tree are c hanged through evolutionary reshaping of genetic trees. Genetic trees are composed of software function nodes, hardware function nodes, and hardware terminal nodes. Software function nodes are respon-

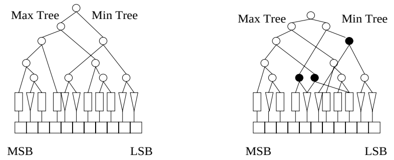

Figure 5: Example

sible for genetic operation. They specify genetic operation probability according to the depth of a node and determine the maximum and minimum value. Hardware function nodes and terminal nodes implement hardware logic to produce the maximum and minimum values to comparator. The values are represented in 10 bits as the output of the GP tree. For hardware efficiency, we use a relatively small population size. Crossover, mutation, and reproduction operators for adapting the shape and size of GP trees are used. Crossover probability is 0.1 and mutation probability is 0.4. As the crossover and mutation probability shows, mutation is more frequently used than crossover. For correct perception, fine tuning of sensor criteria is important. So, mutation becomes the preferred operator. Fitness of programs is measured at each generation by $f = (f_1 + f_2 + f_3)/f_{current}$. Here, f_1, f_2, f_3 denote the number of false classified sensor data of the other sections. $f_{current}$ is the number of correctly classified sensor data of target section. The fitness function represents the ratio of false classification to correctly classified data. As the value of the fitness function becomes closer to 0, more correct interpretation of sensor is possible. For experiment, we selected XC6216 FPGA (field programmable gate array) of XILINX[24]. The FPGA is mounted on H.O.T. Works board from VCC[25]. The configuration is decomposed into part 1 and part 2. Part 1 consists of input registers, 10-bit comparators, and hardware-implemented individuals. At this configuration, each individuals act as a comparison operand. Part 2 is composed of a 12-bit counter and an output register. The comparison and counting tasks are the most time-consuming part of sensor evolution. Because sensor data compared in parallel, the burden of the central processing unit is reduced.

4 Experiment and Result

Two kinds of experiments are performed to verify the proposed method. The experimental environments consist of a single robot and an obstacle. Sensors of a Khepera robot are used for automatic sensor calibration using genetic programming. The goal of experiment 1 is to find the appropriate interpretation criteria for each sensor. In experiment 1, GP runs until the optimal interpretation criteria of one sensor is found. Then, the best individual is used as initial chromosome of searching process for other sensors. The goal of experiment 2 is to find the appropriate interpretation criteria of a sensor under four different conditions. By using GP approach, experiment 2 tries to minimize the overlapped section of four different observed sensor data. Table 2 shows the problem of the experiment 2. In situation 1, the robot rotates to gather sensor data for learning. During rotation, there is no change of the experimental environment. Only the observing sensor is changed as the robot rotates. According to the location angle of four sensor sets, robot rotates four times and records one thousand sensor data at each stop. After rotation, observed data of four sensors are selected and the selected data are used for learning. Using these data and the fitness function, the robot searches the optimal interpretation criteria for each sensor. In experiment 2, the robot does not rotate. The robot starts at its initial location. After data gathering, the robot moves forward for the predefined duration with the fixed velocity and gathers data again. This procedure is repeated four times. Using the gathered data and the fitness function, the robot searches the optimal interpretation criteria which minimize the overlapped section. According to the configuration information of the selected GP individual, the individual section (representing the maximum and minimum value of interpretation criteria) on evolvable hardware is reconfigured. Using this new configuration, the number of data that are interpreted as truth is counted by a counter on evolvable hardware. Using the counted number, the fitness of each individual is computed. Table 3 and table 4 illustrate the experimental results. Table 3 shows the result of the experiment 1. With calibration, the error rate is reduced by maximum 100And in each section, the optimal criterion is found within 20 generations. Table 4 describes the result of the experiment 2. Error rates are reduced by 100

Table 3: Result of experiment 1

Error rate	Section1	Section2	Section3	Section4
Without calibration	2.3586	1.9507	2.7594	2.3496
With calibration	1.8582	0.7873	1.6923	0
Improvement	0.21	0.60	0.39	1.00

Table 4: Result of experiment 2

Error rate	Section1	Section2	Section3	Section4
Without calibration	2.8498	1.2458	1.8375	0.2726
with calibration	2.4054	0.5050	1.4064	0
Improvement	0.16	0.60	0.24	1.00

5 Related Works

According to Brignell, there are two prime motives for the development of smart sensors. The first is to achieve internal compensation of detects. The second prime motive is to take the advantages of developments in digital signal transmission[3]. Evolvable hardware is a reconfigurable hardware whose configuration is under the control of an evolutionary algorithm. Because it can reconfigure itself during runtime, evolvable hardware is able to provide the flexibility of general purpose processor and the performance of ASICs[8, 19]. Therefore, evolvable hardware is suitable for processes that perform repetitive tasks with slight modification of algorithms. Automatic sensor calibration is a good example of such processes. In order to make sensor intelligently, there are two kinds of approaches: sensor optimization and selection of the interested area. It is possible to define that sensor evolution is an approach to construct an optimized sensor topology. Lee introduced a sensor system that learned determination criteria autonomously using genetic programming and neural networks[10]. Menczer reported on experiments using a class of ”latent energy environment” models to define environments of carefully controlled complexity[15]. He studied an agent system that optimized its sensor system against the environment. Lichtensteiger evolved the morphology of an artificial compound eye with 16 light sensors on a robot[12]. Mark optimized eye parameters such as number of agent eye and observation range using PCA and GA[14]. Balakrishnan also performed similar reseach. Balakrishnan optimized the range and placement of the sensors using evolution of neuro-controllers[1]. Sensor evolution research is not limited on the sensor of a single robot. Hackwood used reverse annealing to determine the optimized locations of multiple robots for observing[7]. Selection of the interested area is effort to find more relevant sensory information. Liese studied a model for the evolution of the spectral sensitivity of visual receptors for agents in a virtual environment[13]. Ziegler evolved efficient information processing pathway using artificial chemistry[22]. Nolfi proposed a system that improves its ability by exposing itself only to a sub-class of stimuli to which it knows how to respond efficiently[16]. Cariani introduced an evolutionary method that can obtain epistemic autonomy by adaptively changing perceptual repertoires of sensors[5]. Our work tries to optimize the interpretation criteria by learning. Therefore, it seems that our work belongs to sensor optimization category. But our work corresponds to the base of both approaches. For correct perceiving of the environment, the reliability of a sensor system should be assured. Through automatic sensor calibration, our work aims to improve the reliability of sensors. Usually, intelligent interpretation of sensory data is tried using artificial neural networks[17, 20]. In this work, we chose genetic programming as our learning engine. Genetic programming is a stochastic search method suitable for addressing inductive learning tasks. It uses tree-structured chromosomes inspired by the functional programming of LISP[2, 9]. Therefore, genetic programming can evolve very expressive programs automatically. Furthermore, it is easy to incorporate prior knowledge. Because the computation cost of artificial neural networks is too high for sensor calibration and prior knowledge obtained by other sensor can be used easily, genetic programming is more suitable than artificial neural networks in the approach like ours.

6 Conclusion

We presented a method for sensor calibration using evoklutionary method. This method was motivated by the observation that, with evolutionary calibration, autonomous robots can adapt to their environmental change more easily. We evolved the optimized interpretation criteria for the overlapped sensor data and addressed the problem of misperception by different sensitivity of sensor under the same condition. With evolvable hardware, we could reduce the burden of the central processing unit from repetitive comparisons and counting. Future research can be done in two directions. One is the adaptation to the environment when light sensors are necessary. Another future work includes correctness amelioration by data fusion of a couple of sensors.

Acknowledgements

This research was supported by the Korea Science and Engineering Foundation (KOSEF) under grant 981-0920-107-2.

Bibliography

[1] Balakrishnan, K. and Honavar, V. (1996), On Sensor Evolution in Robotics, *Proceedings of Genetic Programming GP-96*, pp. 455-460.

[2] Banzhaf, W. , Nordin, P., Keller, R. E., and Francore, F. D. (1998), Genetic Programming an Introduction, Morgan Kaufmann.

[3] Brignell, J. E. (1992), The Evolution of Smart Sensors - ASICs the Next Step, *IEE Colloquium on ASICs for Measurement Systems*, pp. 5/1-5/4.

[4] Brooks, R. A. (1989), How to Build Complete Creatures Rather than Isolated Cognitive Simulators, K. Vanlehn (Editor), *Architectures for Intelligence*, pp. 225-239.

[5] Cariani, P. (1998), Epistemic Autonomy through Adaptive Sensing, *Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference*, pp. 718-723.

[6] Celinski, T. and McCarragher, B. (1999), Achieving Efficient Data Fusion Through Integration of Sensory Perception Control and Sensor Fusion, *Proceedings of the 1999 IEEE International Conference on Robotics Automation*, pp. 1960-1965.

[7] Hackwood, S. and Beni, G. (1991), Self-Organizing Sensors by Deterministic Annealing, *IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS 91*, pp. 1177-1183.

[8] Kajitani, I., Hoshino, T., Nishikawa, D., Yokoi, H., Nakaya, S., Yamauchi, T., Inuo, T., Kajihara, N., Iwata, M., Keymeulen, D., and Higuchi, T. (1998), A Gate-Level EHW Chip: Implementing GA Operations and Reconfigurable Hardware on a single LSI,*Proceedings of the Second International Conference on Evolvable Systems*, pp. 1-12.

[9] Koza, J. R. (1992), *Genetic Programming: On the Programming of Computers by Natural Selection*, Cambridge, MA, USA, MIT Press.

[10] Lee, K.-J. (2000), *On-Line Evolution of Complex Robot Behavior Using Neurogenetic Programming*, Master thesis, (in Korean).

[11] Lee, S. and Ro, S. (1997), Uncertainty Self-Management with Perception Net Based Geometric Data Fusion, *Proceedings of the 1007 IEEE International Conference on Robotics and Automation*, pp. 2075-2081.

[12] Lichtensteiger, L. and Eggenberger, P. (1999), Evolving the Morphology of a Compound Eye on a Robot, *Proceedings of the Third European Workshop on Advanced Mobile Robots*, pp. 127-134.

[13] Liese, A., Polani, D., and Uthmann, T. (2000), On the Development of Spectral Properties of Visual Agent Receptors through Evolution, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO 2000*, pp. 857-864.

[14] Mark, A., Polani, D., and Uthmann, T. (1998), A Framework for Sensor Evolution in a Population of Braitenberg Vehicle-like Agents, *Proceedings of Artificial Life VI*, pp. 428-432, MIT Press.

[15] Menczer, F. and Belew, R. K. (1994), Evolving Sensors in Environments of Controlled Complexity, *Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, pp. 210-221.

[16] Nolfi, S. and Parisi, D. (1993), Self-selection of Input Stimuli for Improving Performance, *Neural Networks and Robotics*, pp. 403-418.

[17] Parisi, D., Nolfi, S., and Cecconi, F. (1992), Learning, Behavior, and Evolution, F. Varela and P. Bourgine (Eds), *Toward a Practice of Autonomous Systems*, MIT Press, pp. 845-862.

[18] Rincon, F. and Teres, L. (1998), Reconfigurable Hardware Systems, *Proceedings of Semiconductor Conference CAS98*, pp. 45-54.

[19] Stoica, A., Keymeuler, D., Tawel, R., Salazar-Lazaro, C., and Li, W. (1999), Evolutionary Experiments with a Fine-Grained Reconfigurable Architecture for Analog and Digital CMOS Circuits, *Proceedings of the First NASA/DoD Workshop on Evolvable Hardware*, pp. 76-84.

[20] Thrun, S. (1998), Learning metric-topological maps for indoor mobile robot navigation, *Artificial Intelligence*, Vol. 99, Issue 1, pp. 21-71, Elsevier Science.

[21] Thrun, S. (1996), An Approach to Learning Mobile Robot Navigation, *Robotics and Autonomous Systems*, Vol. 15, pp. 301-309.

[22] Ziegler, J. Banzhaf , W. (2000), Evolving a ”Nose” for a Robot, *Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop Program*, pp. 226-230.

[23] http://kipr.org

[24] http://www.xilinx.com

[25] http://www.vcc.com