Proceedings of the ASME 2010 International Mechanical Engineering Congress and Exposition
IMECE2010
November 12-18, 2010, Vancouver, Canada

IMECE2010-39125

USING BACKPROPAGATION NEURAL NETWORK IN OBJECT RECOGNITION FOR HYBRID MECHANISM MOBILE ROBOT

Samer Charifa
Computer Sciences Corporation
Alexandria, VA, USA

Pinhas Ben-Tzvi
Robotics and Mechatronics Lab
The George Washington University
Washington, DC, USA
bentzvi@gwu.edu

ABSTRACT

Navigation of mobile robots in environments with irregular terrain is a challenging task for scientists and engineers because it involves 3D environment recognition and high order dynamics of the mobile robotic systems. One way to solve this problem is to use hybrid locomotion mechanisms. In this paper, we aim to establish a novel frame work for hybrid local planner which combines motion primitives with artificial neural networks for navigation in rough terrain. This artificial neural network decision engine will find the optimal modes of locomotion for the Hybrid Mechanism Mobile Robot. The input to the neural network is the measurements of the obstruction and the output is the configurations suitable for surmounting an obstacle. Once the robot discovers a change in the terrain (such as high step, low step, etc.), number of measurements will be taken using a feature extraction method to decide the locomotion mode suitable for a particular terrain. Several measurements of the obstruction are considered, such as height, area, and depth of each surface in the scene. These are computed from the 3D representation of the environment built using the on-board sensors, stereo cameras, and a 3D laser range finder. These measurements are fed into a backpropagation neural network in order to choose the successful candidate robot configuration.

Keywords
Mobile robots, Autonomous navigation, Object recognition, Artificial Neural Networks

1. INTRODUCTION

Since Shakey, the first robot to navigate autonomously [1], robots have started sharing the same workspace with humans. Clearly, it is very inefficient to restrict the locomotion of mobile robots to regular wheeled locomotion and consequentially restrict their workspace to indoor environments. However, many problems are involved in rough-terrain navigation such as 3D environment recognition and sometimes motion planning of complex systems with multi-mode locomotion.

A possible approach to reduce the computational complexity of the motion planning in rough terrains is based on restricting the possible trajectories of the robot to a family of curves/straight lines that can be obtained from the interconnection of appropriately defined primitives [2].

Motion primitives and other types of maneuvers have been applied widely to robotics and digital animation. Number of general strategies has been used:

a) Record and playback: This strategy restricts motion to a library of maneuvers. For example, humanoid locomotion can be planned as a sequence of pre-computed steps [3]. Robust helicopter flight can be planned as a sequence of feed-forward control strategies to move between trim states [4-5]. The motion of peg-climbing robots can be planned as a sequence of actions like “grab the nearest peg” [7].

b) Model reduction: This strategy plans overall motion. For example, another way to generate natural-looking humanoid locomotion is to approximate the robot as a cylinder, plan a 2D collision-free path of this cylinder, and follow this path with a fixed gait [8]. A similar method is used to plan the motion of nonholonomic wheeled vehicles [9].

c) Bias inverse kinematic solutions: Like model reduction, this strategy first plans the motion of key points on a robot or digital actor, such as the location of hands or feet or center of gravity. But instead of a fixed controller, a search algorithm is used to compute a pose of the robot at each instant that tracks these points (an inverse kinematic solution). One approach is to choose an inverse kinematic solution according to a probability density function learned from high-quality example motions [10, 11].

Recently, motion-primitive-based algorithms have been successfully used in robotic navigation. Hwangbo et al. [12] have used motion primitives for local motion planning of a single wing UAV. Goldberg et al. [13] have proposed the so called GESTALT navigation algorithm. GESTALT is a set of routines that find the next best direction for a robot to move, given the state of the world already seen, updated sensor data, and a desired waypoint goal. Pivtoraiko et al. [14] proposed a new set of motion primitives which relaxed the condition that all motion primitives are placed on constant curvature arcs. They have shown that by using motion primitives with various curvatures rather than constant curvature will improve the performance of the dual global and local navigation system.

For many robotic platforms, such as indoor mobile navigation or UAV's, the choice of the "best" motion primitive is mostly determined by number of factors such as minimum distance [12], least time, or least energy. For a free-climbing robot, such as the robot considered in this paper, many constraints play equally important role. These are static equilibrium, closed-chain kinematics, collision-avoidance, and torque limits; all of which affect the choice of the winning motion primitive differently at each set of motions to be performed [15].

In this paper, we propose a frame work for hybrid local planner which combines motion primitives with artificial neural networks for navigation in rough terrain.

The application of artificial intelligence techniques such as neural networks in the decision making of the robot has been successfully done in a wide variety of robotic platforms. Antsaklis [16] suggested that the use of Artificial Neural Networks (ANN) in control systems is a natural evolutionary step to meet new challenges because they have the potential for very complicated system. Since then, the ANNs have been used in many aspects of control systems. Robotics motion planning is one of the challenging problems especially for robots with large number of degrees of freedom and eventually higher dimensional configuration space.

Recently, a number of researchers have applied ANN's in robot navigation. Harb et al. [17] have used an ANN to perform object recognition and robot navigation. They were able to recognize environments such as corridors, intersections, corners etc. However, it is not clear how the planning part of their algorithm navigates the robot.

Gao and Han [18] have used neural networks to solve the problem of obstacle avoidance and navigation for an indoor unmanned aerial vehicle based on image data. Pettersson et al. [19] have used neural networks for execution monitoring purposes. They were able to identify model-free failure prediction. Also, Hou et al. [20] have proposed neural network schemes for information processing, localization and navigation of mobile robots. In their approach, they have used the neural network to solve the optimization problem for path planning. The path planning algorithm is based on the minimization of the distance to an obstacle.

The proposed algorithm will use motion primitives to plan the steps for the robot climbing. Moreover, Backpropagation Neural Network (BNN) will be used to choose the winning motion primitive.

2. HYBRID MECHANISM MOBILE ROBOT

Without a loss of generality, the proposed algorithm will be applied to the Hybrid Mechanism Mobile Robot (HMMR) shown Figure 1. The HMMR is a multi-configuration mobile robot, which has the ability to utilize its manipulator arm to climb obstacles as well as for manipulation purposes [21–24]. Furthermore, each link has the ability to be folded inside the previous link so as to change the number of degrees of freedom depending on the required locomotion mode. Consequently, the HMMR can generate several modes of locomotion contingent on whether it fully or partially deploys its manipulator arm. This special property allows it to overcome regular terrains/obstacles (such as stairs, ditches, and steps) and irregular terrains (such as a rubble pile). The full geometrical symmetry of the HMMR provides it with the ability to deploy its links from both sides of the platform, which means that the robot will have the same functionality even if it were to flip over.

Overall, the HMMR incorporates two tracked platforms actuated independently and provide traction to the robot. A central manipulator arm with two links and two degrees of freedom (DOF) is cascaded in between the tracks. The actuation of these DOF's is performed separately via motors located under the tracks.

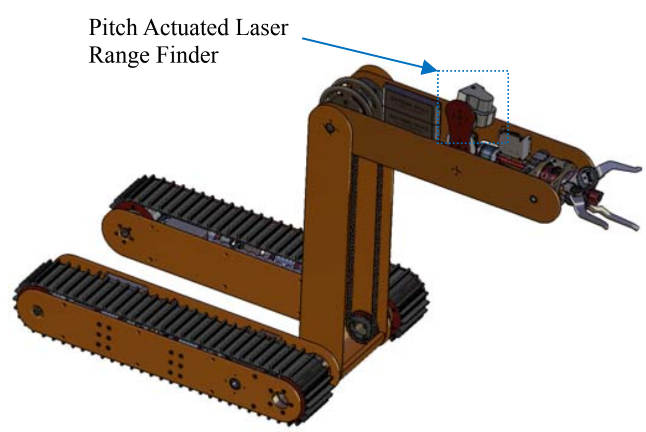


Figure 1: CAD drawing of the HMMR

In addition to the gripper, the third link in the HMMR accommodates a servo-actuated mechanism that carries a Pitch-Actuated Laser Range Finder (PALRF) and a stereo vision system. A single board computer is also housed inside the link. This connects directly to the camera and Lidar in order to process images and synthesize actions accordingly. This robot is currently in the manufacturing process. The detailed description of the HMMR is not in the scope of this paper, but interested readers can find more information in references [21–24]. In this paper, we are interested in presenting the autonomous decision making process of the robot supported by simulations.

3. THE GENERAL FRAME OF LOCAL PLANNING

The dual planning in autonomous navigation is a widely adopted practice in robotics. Most mobile robot navigation techniques are developed at two levels: the local level and the global level. The global planner is assigned the mission of finding the optimal path plan based on the limited prior knowledge of the environment. However, the local planner deals with navigation on the scale of a few meters, where the main problem is obstacle avoidance. The local navigation is mainly useful for rapid responses to avoid collisions. Moreover, the local planner can be of special use for robots which have the ability to surmount an obstruction if needed, like the HMMR. In this case, once the sensors detect an obstruction, the robot should be able to classify this object into classes based on the features of the obstruction. In the HMMR, these features are extracted using the 3D laser range finder. The classification process will not only classify the obstruction into surmountable or non-surmountable obstacles; it should also determine the type of surmounting process that the robot will use. For a reconfigurable robot such as the HMMR, it uses different configurations for different situations. In this paper, these configurations are limited to four types of obstructions: Small step, High step, Stair, Wall. These configurations are just examples to test the classifications process; however it is not limited to the above mentioned four cases. In Figure 2, the flow chart of the proposed local planning algorithm is presented. The algorithm starts with following the global plan. This can be computed using any suitable navigation function such as artificial potential field, Probabilistic Road-Maps, or Rapidly-exploring Random Trees RRTs, etc. In this paper, the harmonic artificial potential field is adopted. The harmonic potential field has no local minima. It performs very well as a global planner. However, it has been shown [25] that the harmonic potential field performs poorly for local planning. In this paper, this drawback is compensated by using the proposed local planner.

The local planner starts once the robot discovers an obstruction using a stereo vision system. Immediately thereafter, the robot will start scanning the environment using Pitch Actuated Laser Range Finder and a 3D cloud (X,Y,Z) representation of the environment will be acquired. This is elaborated upon in the following section. The next step is to divide the 3D cloud into a number of motion primitives in different directions. The motion primitives are shown in Figure 3, where a number of rays are illustrated. Every ray represents a possible direction for a candidate motion primitive. The next step is to compute metrics of these candidates such as height and width and degree of difficulty. These will be fed into a neural network which will label each with a degree of difficulty. For now, we adopted the following levels of difficulty: Wall = 4, stair = 3, high step = 2, low step = 1, flat ground, = 0. The next step will be to find the candidate motion primitives with the lowest difficulty score. If one is found with score 0 then follow it. If none is found then update the number of motion primitives. If multiple are found then check which has the lowest potential field values.

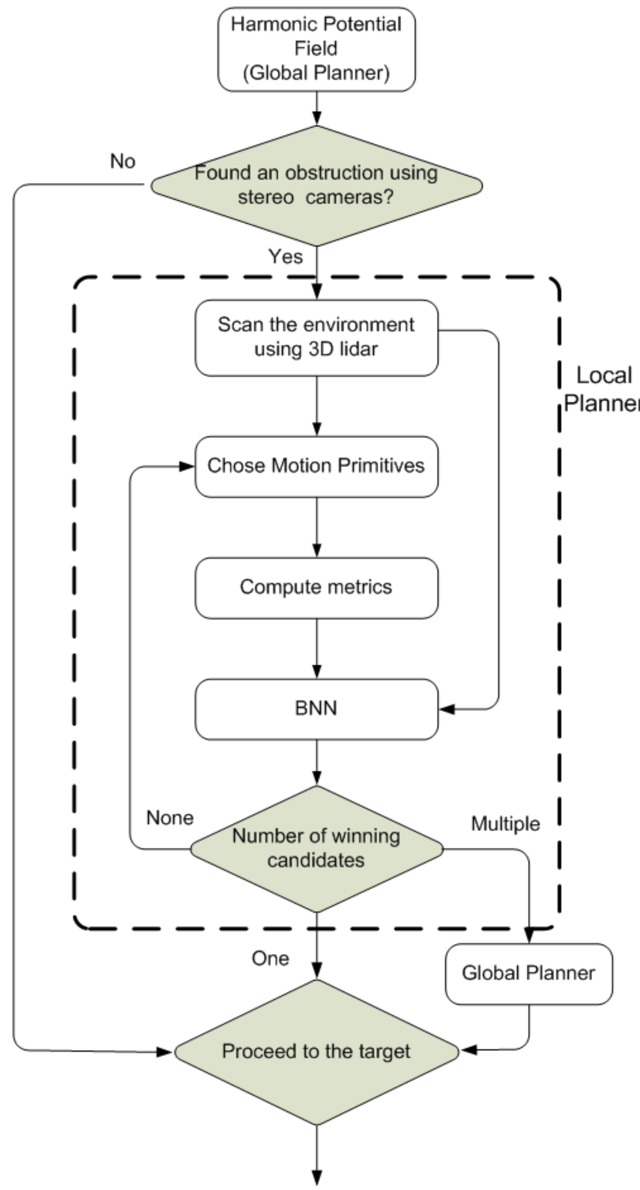


Figure 2: Flow chart represents the general frame of the local planning algorithm

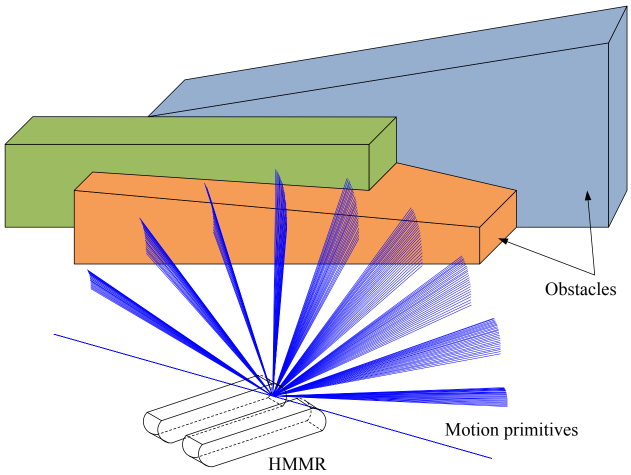


Figure 3: Motion primitives for the HMMR

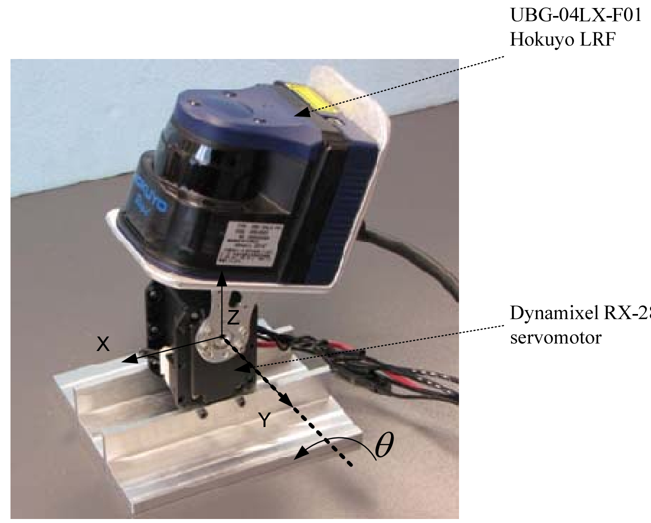


Figure 4: Pitch Actuated Laser Range Finder

4. FEATURE EXTRACTION

In Figure 4, the Pitch Actuated Laser Range Finder (PALRF) is shown. Every point in the point cloud data of the PALRF can be represented in terms of three variables (ρ, ψ, θ), where ρ is the distance from the LRF to the position of the point of interest, θ is the pitch angle, and ψ is the yaw angle. For every pitch angle θ_i, the values of the 2D polar variables (ρ, ψ) readings of the LRF are projected into a local coordinate system (x,y,z). The origin of this coordinate system is located at the center of the scanning level of the LRF. After moving in the pitch direction, all points in the space will be projected into one global coordinate system. We have chosen the global coordinates system (X,Y,Z) to be located at the center of rotation, as shown in Figure 4. The homogenous transformation will transfer any point $P = [P_\rho, P_\psi, P_\theta]^T$ represented using the variables (ρ_i, ψ_i, θ_i), into the global coordinates (P_X, P_Y, P_Z). This transformation is described in the following equation:

$$\begin{bmatrix} P_X \\ P_Y \\ P_Z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_i & -\sin\theta_i & d\sin\theta_i \\ 0 & \sin\theta_i & \cos\theta_i & d\cos\theta_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \rho_i\cos\psi_i \\ \rho_i\sin\psi_i \\ 0 \\ 1 \end{bmatrix}, \qquad (1)$$

where d is the length of the rotating arm. Eq. (1) maps every point in the point cloud into a point in the global coordinate system.

An example of a 3D image generated by the PALRF is shown in Figure 6. Figure 5 shows a 2D image of stairs taken by a digital camera. These stairs were scanned using the PALRF system shown in Figure 4. All the scanned data was projected on one frame using Equation 2. Figure 6 shows a 3D image of the stair using PALRF. A segmentation process has been applied on the 3D image to segment each surface in order to extract useful information from the 3D image. This information may be the height of each step, distance to each step, width of each step, etc. This is shown in Figure 7.

For example, the actual height of each step was 28 cm while the measured height of the steps with the PALRF was around 31 cm. It should be noted here that since the laser scanner measures distance to objects and not heights, the process of projecting these distances into their respective surfaces and subtracting them to get the heights could introduce error. This error may be attributed to two possible factors: the first is the accuracy of finding the edge of the step, and the second may be attributed to hardware limitations of the laser scanner in terms of finding the accurate distance.

Figure 5: Scanning environment of the PALRF

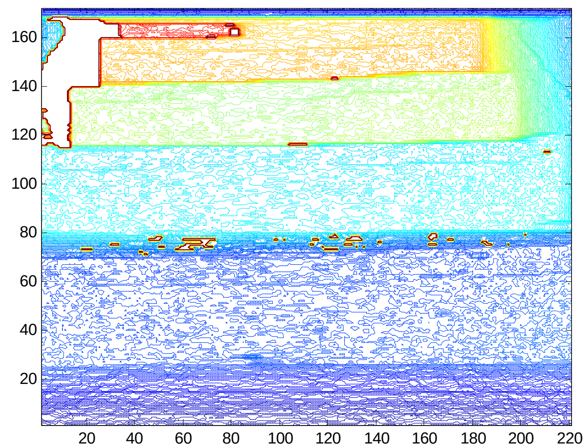


Figure 6: 3D image of the PALRF

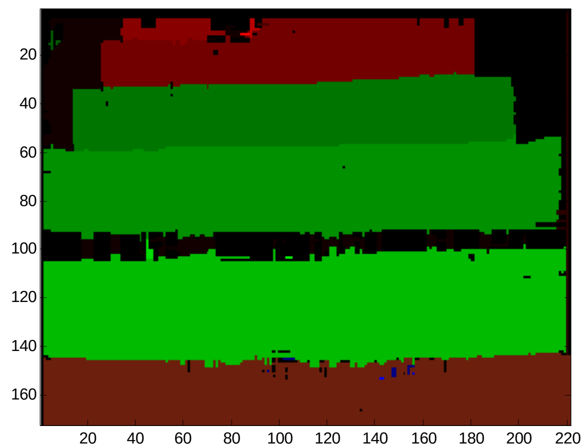


Figure 7: A segmentation of the 3D image into surfaces

5. BACKPROPAGATION NEURAL NETWORK

In this section, the neural network is briefly described. After segmenting the environment, the neural network will input slices of the 3D Lidar image in order to make a decision on what motion primitive the robot should consider, as explained in Figure 2.

We assume that each input event vector X has a dimension of m and each output event vector has a dimension of n (see Figure 7). We also assume that the network has $N+n$ trainable neurons. N can take any value, where $N \geq m$.

The backpropagation neural network can be summarized in the following steps:

- Start with assigning arbitrary values for the weights W
- Next, calculate the outputs $Y(t)$ and the errors $E(t)$ for that set of weights
- Then, calculate the derivatives of $E(t)$ with respect to all of the weights.
- If increasing a given weight would lead to more error, that weight is reduced and vice versa
- After adjusting all the weights up or down, this process is restarted and continued until the weights and the errors settle down

The uniqueness of general backpropagation lies in the method used to calculate the exact derivatives for all of the weights in only one pass through the system.

The GBPN works as two pass: forward evaluations, which use the input events to calculate the network desired output $\hat{Y}$; and Reverse Evaluation Werbos [26] chain rule for ordered derivatives to calculate the error and adjust the weight, as show in Figure 8.

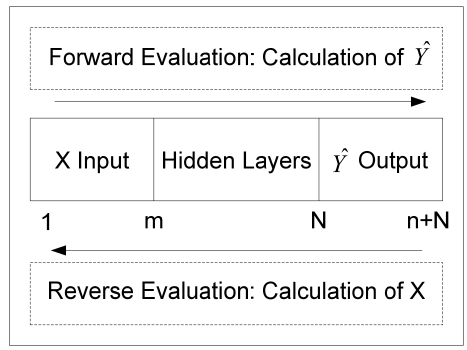


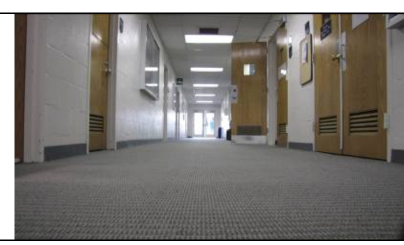
Figure 8: Forward and backward flow of GBPN

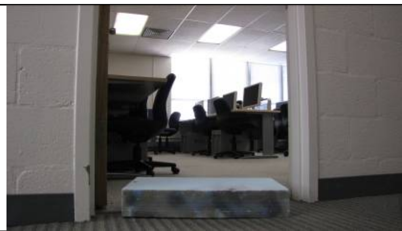
In this paper, the input X is considered to be the set of sub-3D-images, which represents the motion primitives. Each sub-3D-image is a "slice" of the whole 3D image computed by the PALRF. In particular, X is a set of 5x121 matrices. The output vector Y contains the labels of the sub-3D-images.

5.1 Training Data of the Neural Network

In this section, we show samples of training data for the neural network. These training data sets fall into five categories, as shown in Table 1. The HMMR has a library of strategies for each of the five sets. The robot uses different locomotion techniques in dealing with each case. This is explained in the following table.

Table 1: 3D images of different terrains are used as training set for the neural network

Clear: Relatively clear terrain, traversable without changing configuration from normal cruising mode	

Low step: An obstruction that is below the height of the tracks of the HMMR	
High step: A step of sufficient height so as to require more sophisticated climbing techniques	
Stairs: A common special case for a series of low steps	
Wall: An obstacle that is not traversable with any available strategy from the library	

5.2 Testing the BNN

20 cases were collected for each of the above mentioned categories for training. Another set of 10 cases were used for testing. The neural network showed very good classification. The percentages of correct classifications for each case are summarized in the following table.

Table 2: Testing the recognition capability of the BNN-based system

Case	Small Step (SS)	Stair (S)	High Step (HS)	Wall (W)
Number of tested cases	10	10	10	10
Number of missed classifications	0	1	2	1

Note that the testing data varies in shape, distance and orientation from the training data; however, the robot was able to approximate it into one of the known cases.

6. SIMULATION RESULTS

In this section, we have tested the motion primitive algorithm on a scenario that is shown in Figure 9. The local planning algorithm was able to initialize 11 motion primitives in the frontal direction of the robot, as explained in Figure 3. The Motion Primitives (MPs) are input to the BNN and consequentially labeled by their level of difficulty, as shown in Table 3.

Table 3: Labeling results of the MPs shown in Figure 9

Label	W	S	HS
MPs	1,2,3,9,10,11	4	5,6,7

Note that the levels of difficulty are listed from easy to difficult as: flat terrain, small step, high step, stair, and wall. Therefore, according to Table 3, the primitives 4, 5 and 6 will have the easiest path. Considering the configuration of the robot and the global planning, motion primitive 6 would be the winner.

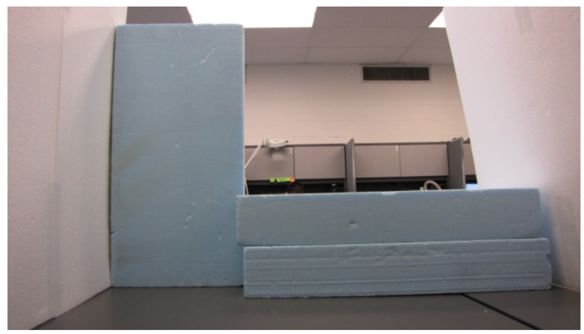


Figure 9: Sample scenario to test the motion-primitive algorithm

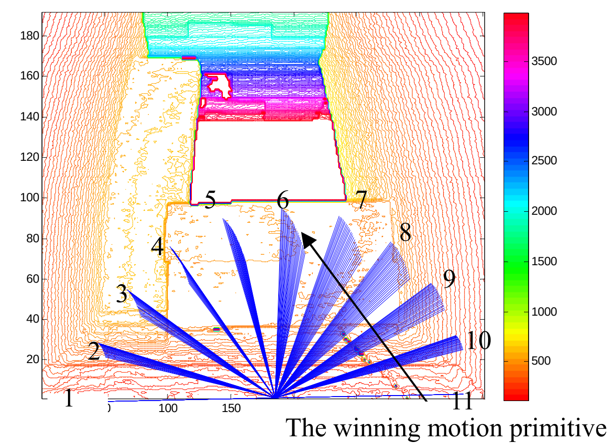


Figure 10: 3D representation of the environment in Figure 9 showing the winning motion primitive (dimensions in color bar are in [mm])

7. CONCLUSIONS

In this paper, we have established a novel framework for local planning applied to climbing robots as well as to robots working in irregular terrains. The novel terrain classification algorithm is accomplished by using motion primitives and a supervised neural network. The feature selection algorithm of the terrain is based on a 3D image of a Pitch Actuated Laser Range Finder. Features of several types of terrains were used to train a backpropagation neural network. The neural network has been tested in a number of environments. The simulation results show correct classification. The output generated was the correct path with the lowest degree of difficulty.

ACKNOWLEDGMENTS

This work is supported by the Defense Advanced Research Projects Agency (DARPA) under grant number HR0011-09-1-0049. We also would like to acknowledge the support provided by the DARPA Program Manager, Ms. Melanie Dumas.

REFERENCES

[1] A Short History of Robots: http://prime.jsc.nasa.gov/ROV/history.html.

[2] Frazzoli, E., Dahleh M.A., and Feron, E., 1999, "Robust hybrid control for autonomous vehicles motion planning", Technical Report LIDS-P-2468, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology.

[3] Kuffner, J.J., Nishiwaki, K., Kagami, S., Inaba, M., Inoue, H., 2003, "Motion planning for humanoid robots", *In International Symposium on Robotics Research*, pp. 1-10.

[4] Frazzoli, E., Dahleh, M.A., Feron, E., 2002, "Maneuver-based motion planning for nonlinear systems with symmetries", *IEEE Transaction on Robotics*, **25(1)**, pp. 116-129.

[5] Frazzoli, E., Dahleh, M.A., Feron, E., 2002, "Real-time motion planning for agile autonomous vehicles", *AIAA Journal of Guidance, Control, and Dynamics*, **25(1)**, pp. 116–129.

[6] Gavrilets, V., Frazzoli, E., Mettler, B., Peidmonte, M., Feron, E., 2001, "Aggressive maneuvering of small autonomous helicopters: A human-centered approach", *International Journal of Robotic Research*, **20(10),** pp. 795–807.

[7] Bevly, D., Farritor, S., Dubowsky, S., 2000, "Action module planning and its application to an experimental climbing robot", *In IEEE International Conference for Robotics and Automation*, pp. 4009-4014.

[8] Ng A.Y., Kim H.J., Jordan M., Sastry, S., 2003, "Autonomous helicopter flight via reinforcement learning", *Processing Advances in Neural Information Systems*, pp. 1-8.

[9] Laumond, J., Jacobs, P., Taix, M., Murray, R., 1994, "A motion planner for nonholonomic mobile robots", *IEEE Transaction on Robotics and Automation*, **10(5)**, pp. 577–593.

[10] Liu, C.K., Hertzmann, A., Popovi´c. Z., 2005, "Learning physics-based motion style with nonlinear inverse optimization", *ACM Transaction on Graphics*, **24(3)**, pp. 1071–1081.

[11] Grochow, K., Martin S.L., Hertzmann, A., Popovi´c, Z., 2004, "Style-based inverse kinematics," *ACM Transaction on Graphics*, **23(3)**, pp. 522–531.

[12] Hwangbo, M., Kuffner J., Kanade T., 2007, "Efficient Two-phase 3D Motion Planning for Small Fixed-wing UAVs", *IEEE International Conference on Robotics and Automation*, pp. 1035-1041.

[13] Goldberg, S.B., Maimone M.W., Matthies, L., 2002, "Stereo Vision and Rover Navigation Software for Planetary Exploration", *Proceedings IEEE Aerospace Conference*, pp. 2025-2036.

[14] Pivtoraiko, M., Nesnas, I., Kelly, A., 2009, "Autonomous Robot Navigation with Advanced Motion Primitives", *IEEE Aerospace Conference*, pp. 1-7.

[15] Britl, T., 2006, "Motion Planning of Multi-Limbed Robots Subject to Equilibrium Constraints: The Free-Climbing Robot Problem", *International Journal of Robotics Research*, **25(4)**, pp. 317–342.

[16] Antsaklis, P.J., 1990, "Neural Networks in Control Systems", Special Issue on Neural Network in Control Systems, *IEEE Control Systems Magazine*, **10(3)**, pp. 3-5.

[17] Harb, M., Abielmona, R., Naji, K., Petriul, E., 2008, "Neural Networks for Environmental Recognition and Navigation of a Mobile Robot", *IEEE International Instrumentation and Measurement Technology Conference*, pp. 1123–1128.

[18] Gao, B., Han, L.S., 2006, "Neural networks implementation of the visual information processing for an intelligent aerial vehicle", *IEEE International Conference on Cognitive Informatics*, pp. 394-399.

[19] Pettersson, O., Karlsson, L., Saffiotti, A., 2007, "Model-Free Execution Monitoring in Behavior-Based Robotics", *IEEE Transaction on Systems, Man, and Cybernetics—PART B*, **37(4)**, pp. 890-901.

[20] Hou, Z.G., Tan, M., Gupta, M.M., Nikiforuk, P.N., Homma N., 2005, "Neural Network Methods for the Localization and Navigation of Mobile Robots", *Canadian Conference on Electrical and Computer Engineering*, pp. 1057-1060.

[21] Ben-Tzvi, P., Goldenberg, A.A., and Zu, J.W., 2008, "Design and Analysis of a Hybrid Mobile Robot Mechanism with Compounded Locomotion and Manipulation Capability", *Journal of Mechanical Design*, 130, pp. 1– 13.

[22] Ben-Tzvi, P., Goldenberg, A.A., and Zu, J.W., 2007, "Implementation of Sensors and Control Paradigm for a Hybrid Mobile Robot Manipulator for Search and Rescue Operations", *In Proc. IEEE International Workshop on Robotic and Sensors Environments (ROSE 2007)*, Ontario, Canada, pp. 92 – 97.

[23] Ben-Tzvi, P., Goldenberg, A.A., and Zu, J.W., 2008, "Design, Simulations and Optimization of a Tracked Mobile Robot Manipulator with Hybrid Locomotion and Manipulation Capabilities", *in Proc. IEEE International Conference on Robotics and Automation, ICRA '08,* Pasadena, CA, pp. 2307 – 2312.

[24] Ben-Tzvi, P**.,** 2010, "Experimental Validation and Field Performance Metrics of a Hybrid Mobile Robot Mechanism", *Journal of Field Robotics*, Vol. 27, Issue 3, pp. 250-267.

[25] Tanner, H.G., Loizou, S.G., Kyriakopoulos, K.J., 2003, "Nonholonomic Navigation and Control of Cooperating Mobile Manipulators", *IEEE Transaction on Robotics and Automation*, **19(1)**, pp. 53-64.

[26] Werbos, P., 1994, The Roots Of Backpropagation: From Ordered Derivatives To Neural Networks And Political Forecasting, Wiley-Interscience, New York, NY.

