# Learning Integrated Perception-Based Speed Control

Pat Leang, Bir Bhanu
University of California, Riverside, CA 92521
pleang@vislab.ucr.edu, bhanu@vislab.ucr.edu

## Abstract

*Advances in the area of autonomous mobile robotics have allowed robots to explore vast and often unknown terrains. This paper presents a particular form of autonomy that allows a robot to autonomously control its speed, based on perception, while traveling on unknown terrain. The robot is equipped with an onboard camera and a 3-axis accelerometer. The method begins by classifying a query image of the terrain immediately before the robot. Classification is based on the Gabor wavelet features. In learning the speed, a genetic algorithm is used to map the Gabor texture features to approximate speed that minimizes changes in accelerations along the three axes from their nominal values. Learning is performed continuously. Experiments are done in real time.*

## 1. Introduction

Robots are usually placed in environments where traversing surfaces are known and predictable. Their wheels are specifically equipped for each surface, and this enables them to drive and maneuver easily through their environment. Some examples include: the smooth marble floors of a commercial building lobby, or a paved road, where the surface is uneven, but even then, familiar and very much known. The purpose of this research is to expand on the familiar kinds of environments upon which current robots are operated by allowing a robot to adjust its speed accordingly to its assessment of the terrain roughness.

Howard and Seraji [1, 2, 3] introduce a traversability index that provides an assessment of the terrain for navigation by measuring, among several other factors, rock size, concentration and terrain slope. The test vehicle can run autonomously run at different speeds depending upon the terrain, however, the speeds are only limited to three types: slow, moderate or fast.

Tunstel, Edwards and Carlson teamed up with Ayanna Howard on the same project [4]. Their strategy uses a fuzzy logic framework for analysis of terrain traversability. The aim is to navigate a robot to a goal via the most easily traversable path at the appropriate speed.

Some robots are used for exploration. Take for example, the Nomad robot, designed at the Robotics Institute at Carnegie Mellon University, was designed to traverse some 200 kilometers across the Atacama Desert in Chile [5, 6]. Also, the famous Mars Pathfinder navigates through the sand dunes of Mars [7]. However, these robots are not entirely autonomous because a human operator controls them remotely.

As compared to the three approaches, the approach presented in this paper involves learning to control speed autonomously based on the images acquired by the robot and the accelerations experienced by the robot. The robot performs continuous learning and captures the mapping of terrain texture features to approximate speeds and organizes its long-term memory to store its experiences over time.

## 2. Technical Approach

The overall technical approach is shown in Figure 1. The robot begins by capturing an image. Clustering is performed on the current feature vectors in the knowledge database to find the closest class. If no close class is found, or if a class is found that does not have the minimum required 10 members, then a new class is made using the features, a randomized speed, and its fitness. If the class does have 10 members, then the robot determines if GA should terminate by calculating whether the standard deviation of the velocities are less than the 2% pulse width modulation duty cycle that drives the robot. If it meets the GA termination requirement, then the robot chooses the best speed within the class based on the fitness values of its members. If GA does not terminate, then crossover is performed on the members to get a new speed value. The process is then repeated with the capturing of a new image.

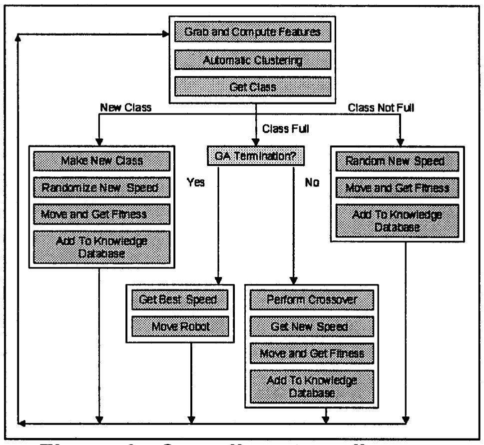


**Figure 1. Overall system diagram.**

## 2.1 Terrain Characterization

Many texture measures exist that can be used for terrain assessment, such as the center symmetric auto-correlation methods presented by Harwood, et. al.[8] and the famous Laws energy texture measures [9]. However, the advantage of the Gabor texture measures is in the fact that it is scale and orientation tunable. We use the Gabor texture [10] features for terrain assessment of a given query image. 2 scales and 4 orientations produce 8 features that double to 16 features by taking the mean and standard deviation. Classification of the query images is governed by an automatic clustering scheme based on scatter matrices.

## 2.2 Genetic Algorithm

Genetic algorithms are used to learn the appropriate speeds associated with each texture feature vector.

A 2-point crossover scheme is used. 2 points are randomly selected as the crossover point. The bits between these 2 points in 1 gene are added to the bits outside the 2 points in the second gene or vice versa.

In mutation, a bit is randomly selected and changed. The bit changes from a 0 to a 1 or vice versa. The mutation rate is set at 1%. GA termination is determined when the standard deviation of speeds corresponding to the 10 best fitness values is less than 2% pulse width modulation duty cycle. This criterion indicates that subsequent crossover will not improve the class by a significant amount and thus, no more crossovers will be done for this class. The fitness is determined by the function,

$$fitness = 1 - \left[ \frac{1}{4}\left|\frac{a_{x,ave} - a_{x,nom}}{a_{x,\max} - a_{x,nom}}\right| + \frac{1}{4}\left|\frac{a_{y,ave} - a_{y,nom}}{a_{y,\max} - a_{y,nom}}\right| + \frac{1}{2}\left|\frac{a_{z,ave} - a_{z,nom}}{a_{z,\max} - a_{z,nom}}\right| \right]$$

$a_{x,ave}$ = average acceleration in the x-direction

$a_{x,\max}$ = maximum acceleration in the x-direction (similar for y and z).

## 2.3 Knowledge Database

The knowledge database is a collection of past experiences [11] that the robot has encountered. It includes the image features, its parameters and the corresponding fitness.

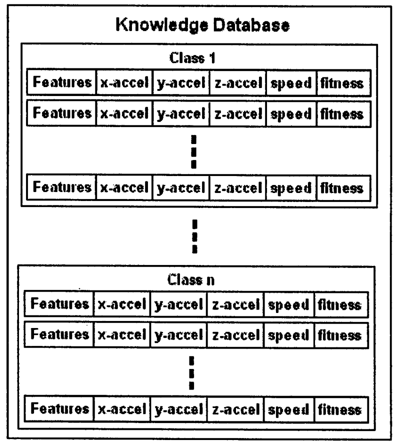


**Figure 2. Knowledge database.**

The knowledge database holds as many items as necessary to represent all the different classes the robot has encountered.

Figure 2 provides an example of the knowledge database. Each class consists of members, where each member has information about a query image and consists of the feature vector, robot accelerations, speed and fitness.

## 2.4 Automatic Clustering

Scatter matrices measure the variance of members within a class and the variance of classes among all classes. These measurements provide information when deciding upon the number of classes that a database should have. In this sense, scatter matrices [12] can help decide on the optimal number of classes given as a set of feature vectors. Given a d-dimensional feature vector and an optimal set of c classes, the set of equations are defined as,

$$\vec{m}_i = \frac{1}{n_i}\sum_{x\in D_i}\vec{x} \qquad \vec{m} = \frac{1}{n}\sum_{\vec{x}}\vec{x} \qquad S_i = \sum_{x\in D_i}(\vec{x}-\vec{m}_i)(\vec{x}-\vec{m}_i)^t$$

$$S_W = \sum_{i=1}^{c} S_i \qquad S_B = \sum_{i=1}^{c} n_i(\vec{m}_i-\vec{m})(\vec{m}_i-\vec{m})^t$$

$n$ is the total number of members in all classes, $n_i$ is the number of members in class $I$, $D$ is the entire dataset, in which $D_i$ is the dataset for class $I$, $m$ is the mean of the entire dataset and $m_i$ is the mean for class $I$, and $t$ denotes the transpose of a matrix.

The within-class scatter measures the separability of the members of a class and the between-class scatter measures the separability of the classes. In the optimal case, the within-class scatter should be small and the between-class scatter should be large. This will ensure that the members of each class cluster close to its mean and the centers of all clusters are well separated. The technique normally used is to maximize the equation,

$$Beta = \frac{Trace(S_B)}{Trace(S_W)}$$

The optimal number of classes occurs when the value of *Beta* is at its maximum.

Members of the dataset are classified using the K-means algorithm. Members are classified into the class for which its distance to the class mean is the minimum. Initially, all members are classified into a single class. The number of classes increases until its subsequent *Beta* value is less than its previous value.

## 3 Experiments

The experiments are done in real time on a mobile robot. The robot is equipped with an onboard camera.

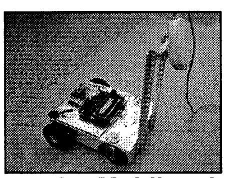

**Figure 3. Mobile robot.**

The robot also has a 3-axis accelerometer, allowing it to calculate accelerations in the x, y and z directions. These acceleration values are used for the calculation of the fitness values for the genetic algorithm. Figure 3 shows a picture of the robot.

The textures were synthetically fabricated with various materials. Three texture materials are shown in Figure 4.

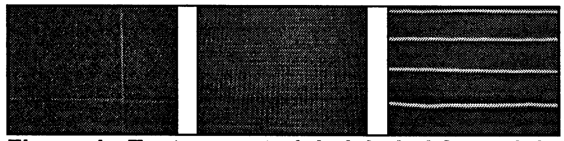

**Figure 4. Texture materials labeled from right to left: sand paper, steel wire grid, and clothesline wire.**

### 3.1 Iterative Robot Motion Description

Each iteration begins with the capturing of an image and computing its Gabor Featues, and ends when the appropriate speed is determined. The robot will move, collect accelerometer data and compute a fitness value. All information collected up to this point will form the new member that is added to the knowledge database.

The standard deviation of the speeds of the 10 best members is determined ranked according to their fitness values. If the standard deviation is not less than a particular threshold, then crossover will be performed using the 10 best members. From the offspring, the robot will drive at the new speed, obtain its fitness, and form a new member that is added to the knowledge database.

If the standard deviation is less than the 2% threshold, then GA terminates for that particular class, and the optimum speed will be the speed value of the member with the highest fitness in that class.

### 3.2 Final Results

Examples are given for three terrain types made from sand paper, steel grid wires and clothesline wires. Figure 5 shows the fitness as the speed increases for terrain type labeled as "Sand Paper". Similar results appear for "Steel Grid" and "Clothesline" in which their optimum speeds are reached at 56% and 20% PWM duty cycles. 20% to 80% duty cycles corresponds linearly to 3.3 to 5.5 in/sec.

Figures 6 through 8 show examples when GA has terminated for the terrain type. The figures show that convergence is encountered after 20 to 35 iterations.

The final speeds differ by only 2% duty cycle as compared to its actual optimal value. Final number of classes converges to 3, 4, and 1 for sand paper, steel grid, and clothesline wire.

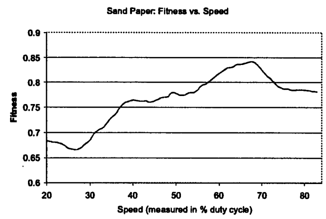


**Figure 5. Fitness vs. speed for "Sand Paper."**

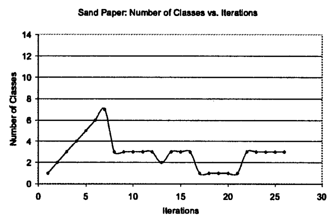


**Figure 6. GA with clustering for "Sand Paper".**

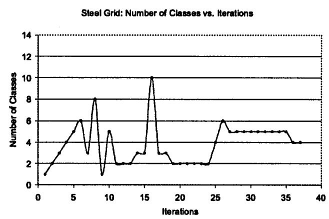


**Figure 7. GA with clustering for "Steel Grid".**

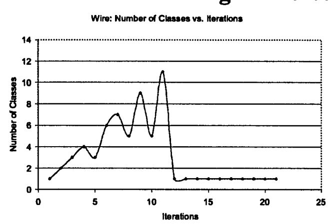


**Figure 8. GA with clustering for "Clothesline".**

A total of 5 different terrain types were experimented upon that produced a total of 13 classes. An average of 30 iterations was required to converge upon the appropriate speed.

## 4. Conclusions

A complete automatic speed control for a mobile robot is successfully implemented using several techniques that include the Gabor feature extraction method, genetic algorithms, the knowledge database, scatter matrix clustering, and the K-Means algorithm. The experiments are tested with real data on a mobile robot. They demonstrate that a robot can successfully learn the appropriate speed in unknown terrains, in a completely autonomous manner. This research has been successful on multiple texture types. Further experiments include testing on real world terrains and in much larger environments.

## References


[1] A. Howard and H. Seraji, Real-Time Assessment of Terrain Traversability for Autonomous Rover Navigation. *Proc. IEEE/RSJ International Conf. on Intelligent Robots and Systems,* Vol.1, pp.58-63, 2000.

[2] A. Howard, H. Seraji and E. Tunstel, A Rule-Based Fuzzy Traversability Index for Mobile Robot Navigation. *Proc. ICRA,* Vol.3, pp.3067-71, 2001.

[3] A. Howard and H. Seraji, A Real-Time Autonomous Rover Navigation System, World, *Automation Congress*, June 2000.

[4] A. Howard, et. al., Enhancing Fuzzy Robot Navigation Systems by Mimicking Human Visual Perception of Natural Terrain Traversability, *Proc. Joint 9th IFSA World Congress and 20th NAFIPS International Conf.,* Vol.1, pp.7-12, 2001.

[5] W.L. Whittaker, D. Bapna, M. Maimone and E. Rollins, Atacama Desert Trek: A Planetary Analog Field Experiment, *i-SAIRAS'97*, 1997.

[6] D. Bapna, et al., The Atacama Desert Trek: Outcomes. *Proc. ICRA*, pp. 597-604, 1998.

[7] A. Mishkin, et al., Experiences with Operations and Autonomy of the Mars Pathfinder Microrover, *Proc. IEEE Aerospace Conf.,* Aspen, CO, 1998.

[8] D. Harwood, et al., Texture Classification by Center-Symmetric Auto-Correlation using Kullback Discrimination of Distributions, *Pattern Recognition Letters*, Vol. 16, No. 11, pp. 1-10, 1995.

[9] L. G. Shapiro and G. C. Stockman, Computer Vision, Prentice Hall, 2001.

[10] B.S. Manjunath and W. Y. Ma, Texture Features for Browsing and Retrieval of Image Data, *IEEE Trans. PAMI*, Vol.18, No.8, pp.837-42, Aug. 1996.

[11] B. Bhanu, S. Lee, Genetic Learning for Adaptive Image Segmentation, Kluwer Academic Publishers, Jan. 1994.

[12] R. Duda, et al., Pattern Classification, Wiley, John & Sons, Incorporated, Dec. 1999.