

Efficient Roughness Recognition for Velocity Updating by Wheeled-Robots Navigation

Farid García & Matías Alvarado

Centro de Investigación y de Estudios Avanzados – IPN, Departamento de Computación,
Av. Instituto Politécnico Nacional 2508, San Pedro Zacatenco, CP 07360, México DF
farid@computacion.cs.cinvestav.mx, matias@cs.cinvestav.mx

Abstract. In this paper is shown that the Appearance-Based modeling is the best pattern recognition method for supporting the velocity updating of wheeled-robots navigation. Although Appearance-Based recognition algorithms have lower accuracy than the ones for detailed pattern recognition, they successfully classify terrain textures by regarding the average of the appearance. Actually, the detailed recognition algorithms success in recognizing patterns depicted with lines, dots or borders, but they fail for recognizing patterns where the average appearance is required. As human driving experience shows, the assessment of the average appearance is needed for velocity updating during navigation on outdoor terrains. Human drivers make the velocity adjusting based on an estimation of the terrain average appearance. Hence, as the experimental result illustrate, the algorithms for average appearance recognition are the best option for training wheeled-robot for velocity updating while navigating over outdoor terrains.

Keywords: Roughness Recognition, Velocity Updating, Wheeled-Robots Navigation.

1 Introduction

Outdoor autonomous robots are relevant for terrain exploration missions. The terrain difficulties of solar system planets –like Mars–, to move through terrains having soil, rocks and slopes, requires the usage of robots with the highest degree of autonomy to overcome such difficulties [1]. As well, in Earth exploration missions where human lives may be in dangerous circumstances, the autonomous robots are as well required. For instance, search of landmines or exploration of active volcano craters. Autonomous navigation on outdoor terrains is highly complex, obstacle detection and avoidance as well as the terrain features information for no slides, are both required. Environment data must be accurate and quickly processed by the robot's navigation systems. Besides, when data from human remote controllers is not quickly available, the autonomous robots should be equipped for convenient reactions, particularly in front of unpredicted circumstances. Actually, beyond the obstacle location and avoidance, the robot's velocity control, regarding the terrain features, has been few attended and it is a weakness for efficient and safe navigation nowadays.

The classification of terrain roughness has just recently been a bit more attended [2]. In [2] a path over a rough terrain is generated with a terrain-based criterion function, and then the robot is controlled so as to move on the chosen path. In [3] the navigation strategy assesses the terrain's features of roughness, slopes and discontinuity. Larson et al. [4] analyze the terrain roughness by means of spatial discrimination which then is (meta-) classified. In [5] roughness recognition is by using artificial vision, so novel textures recognition is later to an off-line recognition training from sample texture. Pereira et al. [6] plotted maps of terrains incorporating roughness information that is based on the measurement of vibrations occurring in the suspension of the vehicle; this online method can recognize textures at the moment the vehicle passes over them, what is a limitation for remote recognition.

For the purpose of velocity updating for autonomous navigation on rough terrains we claim that is not required to identify textures at high-detail level. Actually, as analyzed in Section 3, high precision recognition methods like Local Binary Patterns (LBP) [7], or Advanced Local Binary Patterns with Rotation Invariance (ALBPRI) [8], having 97.54% and 99.64% of respective efficacy, do not well performed accounting recognition of outdoors terrains textures. The listed accuracy percentages correspond to tests carried out on the texture image database of the Columbia Utrecht Reflectance and Texture Database [9], which is the most common benchmark used for testing texture recognition algorithms.

The LBP and ALBPRI methods have good performance for texture recognition. But these works do not mention anything about recognition of new textures, that is, nothing is said about how a different texture from the texture training set is classified. They just verify if the testing textures belong to any class of the texture training set, i.e., they only give two result values, false and true.

For our purpose, we desire to determine how similar the set of test textures and the set of training textures they are. The Appearance Based Vision (ABV) [10] method having 75% of detailed texture recognition efficacy is good enough for the velocity updating during outdoor navigation as results show in Section 3.2. Although ABV does not take into account the fine details of textures, it captures the so called average appearance of the textures. In other words, with a testing texture, even if it is a new texture, the ABV method compares it with the classes of the training set and indicates the texture class that resembles more to them, according to the average appearance.

In this paper is proposed to improve the process of robot velocity adaptation, by regarding the terrain features and imitating as human beings do. Humans use a quick imprecise estimation of the terrain features but enough to navigate without slides or falls. The human's estimation on the right velocity to safe navigate on irregular terrains is via imprecise but enough surface texture recognition [3]. Actually, we show below that concerning terrains exploration for robot navigation, the highest precision methods for texture recognition are not the adequate but failed.

Surface textures are captured via artificial vision, after image processing the estimation of texture class is gotten as well as the slopes inclinations. The algorithm's output indicates the velocity the robot can move depending on the terrain features. Bright and uniform lighting during navigation is required to guaranty consistent roughness recognition; therefore the presence of shadows, which treatment is a hard task to pattern recognition [11] is out of the scope of this work.

The rest of the article is organized as follows: Section 2 summarizes the closest antecedents in the field of texture recognition; then, the method and architecture of the fuzzy neural network for velocity updating is introduced. Section 3 describes tests and experimental results. A brief discussion is in Section 4, and the paper ends with conclusions.

2 Outdoor Terrains Recognition

Texture recognition is an issue that has been studied extensively; the local binary pattern-based methods are widely used for its good performance in the recognition of textures. Textures are modeled with multiple histograms of micro-textons; the micro-textons are extracted with a local binary pattern operator.

LBP [7] is a gray-scale invariant texture primitive statistic. For each pixel in an image, a binary code is produced by thresholding its neighborhood with the value of the center pixel. A histogram is created to collect up the occurrences of different binary patterns. LBP can be regarded as a micro-texton operator. At each pixel, it detects the best matching local binary pattern representing different types of curved edges, spots, flat areas, etc. After scanning the whole image to be analyzed, each pixel will have a label corresponding to one texton in the vocabulary. The histogram of labels computed over a region is then used for texture description.

Conventional LBP just considers the uniform patterns in the images. It discards important pattern information for images whose dominant patterns are not uniform patterns. ALBPRI [8] proposes a new rotation and histogram equalization invariant texture classification method by extending the conventional LBP approach to reflect the dominant pattern information contained in the texture images and capturing the spatial distribution information of dominant patterns.

These methods do well recognize patterns depicted with lines, dots or borders, but fails for recognizing different depicted appearances. When human drivers drive vehicles, they do not inspect the terrain textures with a magnifying glass nor take a look at a small distance to account details of particular lines, dots or borders; they just estimate the texture roughness basing on previous pattern recognition experience while driving [3].

Human beings classify textures according to past experience; when human drivers find a novel terrain texture, they employ their experience to estimate how rough the novel texture is. Then, they decide how fast they can drive without slide risks. By using fuzzy logic the human process for identifying the terrain roughness can be modeled in such a way to be used by the robot mimicking this human ability. To imitate the human experience during terrain recognition for navigation, it is clever to pay attention in convenient methods for recognizing the surface appearance average, and such that not lost in unnecessary details for the outdoors navigation purpose. Moreover, frequently the surfaces details recognition is computationally high-cost and it should be avoided for autonomous navigation. Actually, the Appearance Based Vision method is good enough for the velocity updating during outdoor navigation.

2.1 Appearance-Based Vision

The ABV method gets the principal components of image distribution, namely, the eigenvectors of the covariance matrix of the object images set. The ordered eigenvectors fashion the features accounting and characterizing the variation among the different images. The use of ABV for object recognition involves the next operators and operations.

Let $\{\mathbf{I}_1, \dots, \mathbf{I}_N\} \subset \mathbf{R}^{n \times m}$ the set of training images, all the images are stacked so that we obtain the set $\{\phi_1, \dots, \phi_N\} \subset \mathbf{R}^{n \cdot m}$. The vectors are normalized with $\tilde{\phi}_i = \phi_i / \|\phi_i\|$. The average vector is computed, $\mathbf{C} = \frac{1}{N} \sum_{i=1}^N \tilde{\phi}_i$. The images are centered by subtracting the average vector with each image; resulting vectors are placed to form the matrix $\Phi = [\tilde{\phi}_1 - \mathbf{C}, \dots, \tilde{\phi}_N - \mathbf{C}]$. The covariance matrix is computed, $\Omega = \Phi \Phi^T$ and its eigenvalues and eigenvectors are calculated. The eigenvectors are ordered in a decreasing fashion according to the eigenvalues, where they are placed as columns of matrix Ψ . All the training images are projected to the eigenspace with $\theta_i = \Psi^T (\tilde{\phi}_i - \mathbf{C}), i = 1, \dots, N$.

In the recognition phase the testing image is projected into the eigenspace and a supervised neural network classifies the image. In other words, let \mathbf{I}_t the testing image, it is stacked and normalized, $\tilde{\phi}_t$. Then, it is projected to the eigenspace with $\omega_t = \Psi^T (\tilde{\phi}_t - \mathbf{C})$. A supervised neural network classifies ω_t .

During outdoors navigation, human drivers estimate the convenient vehicle velocity by regarding their previous experience when driving on similar terrain textures. In other words, human drivers estimate how rough, in average, the terrain is, instead if specific texture details are recognized. Human drivers that navigate on uneven terrains do not need to learn, or to know, about specific details but on the textures appearance average. The average recognition of textures, as the humans do, is the behavior that is mimicked and implemented in order to strengthen the robot navigation abilities.

2.2 The Fuzzy Neural Network for Velocity Updating

For robot velocity updating according to the terrain features, our proposal sets to imitate as human beings do. For safe navigation on irregular terrains, the human's velocity estimation is via imprecise but enough surface texture recognition [3]. When a human driver observes a novel terrain texture, uses his experience to estimate how rough the texture is; then decides the convenient car driving velocity. Thus, in the first step, the terrain's textures are neural-net-clustered in a roughness meta-class: a Supervised Neural Network (SNN) classifies textures; then, a Fuzzy Neural Network (FNN) makes a roughness meta-classification from the terrain texture class. By adding the texture roughness setting and the slope data, the FNN matches each terrain roughness with the corresponding velocity meanwhile the robot navigates safely.

For detection of slopes inclination, an infrared sensor located in the frontal part of the robot does parallel ray projection to the robot's motion; the other sensor projects its ray directly to the floor perpendicular to the first sensor. The inclination angle of slopes is computed by trigonometric operations. The off-line and on-line steps to update velocity regarding the terrain roughness and the inclination slopes while navigating are next described:

Off-line training steps

- 1) Select and model the representative outdoor textures images of the robot's environment. The images are captured while the robot is stationary.
- 2) Train the SNN to learn the texture classification established by the human expert driver.
- 3) Train the FNN to determine the velocity regarding the texture classes as well as the inclination angle of slopes, according to an expert driver's directives (build the fuzzy sets and make the inference IF-THEN rules system).

On-line steps

- 4) Acquisition of terrain images from the robot while it is in motion.
- 5) The SNN classifies the texture, this information is forwarded to the FNN.
- 6) The FNN inputs are the texture class and the slope inclination angle. The FNN indicates the updated velocity to the robot mechanical control system.
- 7) The cycle is repeated as the robot moves, and the velocity is cycle updated.

Following is the architecture of the five-layer FNN. The terrain features recognition followed by the robot velocity tuning is as shown in Fig. 1.

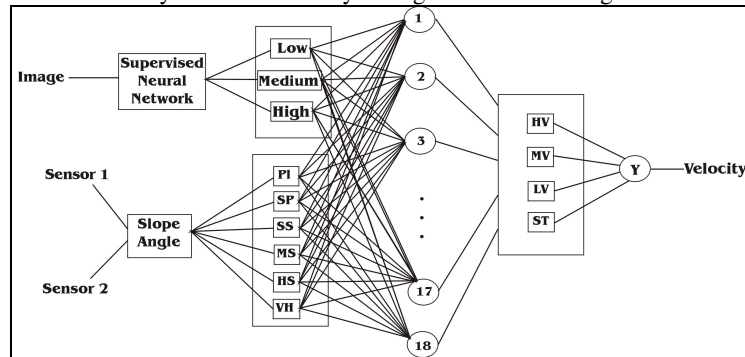


Fig. 1 The Fuzzy Neural Network

The texture class and slope input data are assessed to adjust the velocity that is the FNN output data. The FNN first layer inputs are the slope size and the texture class, the second layer sets the terms of input membership variables, the third sets the terms of the rule base, the fourth sets the term of output membership variables, and in the fifth one, the output is the robot's velocity. The textures roughness is meta-classified in three fuzzy sets, High (H), Medium (M) and Low (L). The inclination angles of slopes are meta-classified in six fuzzy sets: Plain (PI), Slightly Plain (SP), Slightly Sloped (SS), Moderato Sloped (MS), High Slope (HS) and Very High (VH). The FNN output values are either: High Velocity (HV), Moderate Velocity (MV), Low

Velocity (LW) or Stop (ST). Membership functions of the input and output variables denote the corresponding texture roughness, slope inclination angle and velocity, respectively. The FNN output sets the velocity the robot can move safely.

The fuzzy-making procedure maps the crisp input values to the linguistic fuzzy terms with membership values in $[0,1]$. In this work the trapezoid membership functions (MF) for texture variable and the triangle MF for angle variable are respectively used. Taking X, Y, Z as variables of the respective predicates, the general form of inference rules is:

IF *Slope angle* is X **AND** *Roughness* is Y **THEN** *Velocity* is Z.

The inputs parameters are the slope angle and roughness, and the output is the estimated velocity. The de-fuzzy procedure maps the fuzzy output from the inference mechanism to a crisp signal. When the robot finds a slope steeper than the allowed threshold, it stops, and evaluates which movement to make, whose decision concerns to path planning.

3 Experimental Steps

A car-like Bioloid robot transformer kit [12] is used, which uses a processing unit, four servomotors for power transmission to the wheels, two infrared sensors located in the robot front, and a wireless camera on top-front of the robot. The robot dimensions are 9.5 cm width per 15 cm length. In these experiments the SNN is trained with terrain textures from images in Fig. 2. In this platform it is used a personal computer (PC) and the processor of the robot, to form a master-slave architecture, communicated wirelessly. On the PC is implemented and executed the velocity estimation algorithm. The robot, on one hand, reports to the PC the sensors readings and wirelessly transmits the images captured by video camera, on the other hand updates the velocity in accordance with instructions that the PC communicates it.



Fig. 2 Robot's navigation on outdoor surface

The navigation tests focused on velocity updating, the vehicle navigates on the terrain shown in Fig. 2; the robot recognizes textures from the captured images. As soon as significant change of terrain textures is detected the PC indicates the robot to update its velocity according to the terrain texture currently recognized.

In here reported experiments textures are respectively modeled with ABV, LBP, ALBPRI and a two-dimensional Fourier transform method for roughness

classification of cast surfaces (FCS) [13], which it is deserved to assess the cast surface quality. Terrain images are used during the robot’s training for texture recognition of the surfaces it navigates. There were conducted 15 tests with each method, the texture images are garden ground covered by a thin layer of dust, and dispersed little rocks; the grass is 2-centimeter cut height and dry; the paving stone contains leaves, tree branches with a thin dust cover.

3.1 Algorithm’s Performance

The expected robot behavior is that the robot increases its velocity, from low to high, when it is detected grass, ground and paving stone. That is, the robot must move slowly on grass, fast on paving stone; on ground, faster than in grass but slower than in paving stone. The robot’s navigation velocity updating results, regarding the surface textures, are displayed in Table 1, whose data are average velocity values. It shows that the robot moves faster on paving stone than on grass, and in turns, faster on ground than on grass.

Table 1 Velocity results in centimeters per second units

Method/Texture	Ground	Grass	Paving Stone
ABV	9.03	4.4	11.43
ALBPRI	9.68	7.13	11.7
LBP	12.18	11.7	12.85
FCS	1.08	4.3	2.16

The worst performances were with the FCS and LBP methods, the robot did not adjust its velocity as expected. The methods ABV and ALBPRI had better performance, because the robot adjusted its velocity according to the expected behavior. However, with ALBPRI, the estimated velocity for grass is relatively high, because it is close to ground value. With ABV, the velocities are slightly lower than those of ALBPRI, but ABV estimates a lower velocity for grass.

During the training attempts using images from a low roughness texture wall, in addition to other textures occurs that both, the ALBPRI and LBP methods do miss to identify the diverse surface textures, hence misclassifying all of them as wall texture. A likely explanation is that both recognition methods are based on the detection of borders, edges and dots in order to guaranty well recognition performance. Because the wall images are uniformly plain, the required graphical elements for methods well performance are not present. It provokes the methods losing to recognize surfaces not having clearly marked lines, borders or dots. However, wall images were replaced by another kind of wall images with a certain line pattern.

On the other hand, the ABV method does properly recognize the surface textures changes, and then the robot’s velocity is updated, while it is displacing, according to the surface features. Therefore, experimental conclusion is that ABV advantages to LBP and ALBPRI to model and recognize the physical average appearance of textures. By using ABV the recognition of, what we call, the texture average among images is competent. ABV method well recognizes the wall plain texture average, as well as the texture average of grass, ground or paved. Thus, the textures average recognition supports the robots navigation on outdoors terrains. Car drivers do speed

update by regarding the terrain texture average, i.e., velocity adjustments are according to the terrain appearance average variations, which are relevant to human drivers by navigating, and disregarding the irrelevant specific lines, dots or borders for navigation.

The FCS method shows high performance in classification by recognizing polished surfaces like glass, steel or plastics. The method applied to terrain images failed hardly. All the results were wrong; the method misclassified the terrain textures as low roughness textures. The plausible explanation is that this method works on polished textures that require high precision during recognition. But it misses on rough surfaces like grass, ground, soil or pave that do not demand a high precision during textures recognizing.

3.2 Simulation of Real Car Navigation

In this set of tests, images of the terrain textures are taken from a video film recorded by a video camera placed two meters above the floor on a car's roof. The camera recorded the car's path, under a visual field similar to that of a human driving a real car, see Fig. 3.

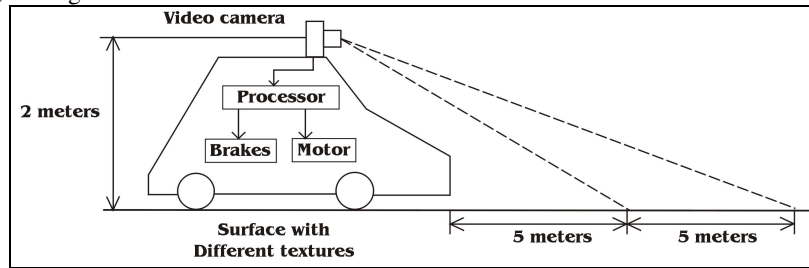


Fig. 3 Car vision/recognition system

The outdoor terrains textures images are loose stones, ground with grass, ground, asphalt and concrete, on which the truck was moving through. These images are used to train the SNN by using the ABV model for textures treatment. The experience of a human drive allows for defining the classes of textures and the respective car velocity. Tests of simulated car navigation are regarding that the vehicle maximum speed it can reach is 50 km/hr. The velocities are as follow: on loose stones, velocity is smaller than on ground with grass, and it is smaller than on sole ground, and in turns, it is smaller than on paved ways; these results are shown in Table 2, showing the average of the velocities resulting in specific experiments.

Table 2 Velocity updating results

Texture	Velocity km/hr
Loose stones	10.65
Ground with grass	18.47
Ground	27
Concrete	43.79
Asphalt	48.91

The minimum and maximum velocities recorded for loose stones are 10.06 km/hr and 18.91 km/hr, respectively. The velocities estimated for ground with grass are a little higher. Mostly the velocity remains at 18.11 km/hr. Ground with grass texture is less abrupt than loose stones. The grass does not cover the entire surface but there are holes with ground, usually small, even so a car can overcome them at low-speed motion by the time it avoids damaging vibrations in the vehicle. Even when grass is a texture that favors slipping and skidding, but unlikely the loose stones, the wheels surfaces of the car have better contact with ground, thus the risk of skidding is smaller, but higher than in the next textures. For ground textures, the velocity remains constant in almost the entire path at 27.61 km/hr. The surface of ground texture is covered with dust and very small stones, and is almost flat, so the vehicle can move fast without being affected by strong vibrations, even the small stones and dust in the surface could make the car to skid. Velocity for concrete texture remains constant, 44.44 km/hr in almost the entire path. Finally, the velocities estimated for the asphalt remain without changes, 48.91 km/hr throughout the entire path. The textures of concrete, paving city streets, and the asphalt covering roads are very similar. The covering with this matter create a uniform surface, without holes and slopes, so avoiding the car skids.

An additional aspect to consider concerns with the vehicle's computing capacities for processing the texture images and the velocity updating in real time. Actually, 1) determine the range of the camera to capture images of the surface, and 2) the sampling time given the progress of the vehicle. The acquired images have a resolution of 480x640 pixels in grayscale. The microprocessor employed was a Centrino Core 2 Duo at 2GHz and 1.99Gb RAM. The processor spends 0.3 seconds for both image processing and velocity updating.

Actually, for efficient velocity control it must be considered the 0.3 seconds the process spends for texture recognition and velocity updating, by assuming that the maximum speed is around 50 km/hr, hence the vehicle will advance 5 meters. As shown in Fig. 3, the camera must process the next 5-meter road segment before the vehicle passes on. That is, when the vehicle moves the first 5-meter stretch, the computer processes the image of the posterior 5-meter stretch. When the second stretch processing is finished, the vehicle would have started to move in the second stretch. This cycle is successively repeated.

4 Discussion

Velocity updating according to the surface roughness, is a subject that has not been fully addressed. Most of the works focus on the detection and obstacle avoidance problem. For instance, Labert et al. [14] use a probabilistic modeling to avoid or to mitigate eventual collisions, regarding the environment perception, by updating a robot braking action. Selekwa et al. [15] and Ward & Zelinsky [16] addressed the navigation and path planning of an autonomous robot which varies the velocity according to the proximity of obstacles detected by infrared sensors. So far, all the referred works on outdoors autonomous robots do not include in their proposals information from terrain surface roughness during navigation.

Conclusions. The efficiency of algorithms for recognition of roughness textures is the key point for allowing velocity updating. According to results the appearance average instead of high-detailed recognition is the requisite for velocity updating on rough terrains. A clever issue is the human mimicking about the recognition and decision making for velocity updating. Human drivers make quick terrain recognition but enough to a right speed updating during navigation. The computationally low-cost and easy implementation of the algorithms make this approach suitable for velocity updating of wheeled-robots during autonomous navigation on outdoor terrains.

References

1. Bajracharya, M., Maimone, M.W., Helmick, D.: Autonomy for Mars Rovers: Past, Present, and Future. *Computer* 41, 44-50 (2008).
2. Ishigami, G., Nagatani, K., Yoshida, K.: Path Planning for Planetary Exploration Rovers and Its Evaluation Based on Wheel Slip Dynamics. In: *IEEE International Conference on Robotics and Automation*, pp. 2361-2366 (2007).
3. Seraji, H., Howard, A.: Behavior-Based Robot Navigation on Challenging Terrain: A Fuzzy Logic Approach. *IEEE Trans Robot Autom* 18, 308-321 (2002).
4. Larson, A.C., Voyles, R.M., Demir, G.K.: Terrain Classification Using Weakly-Structured Vehicle/Terrain Interaction. *Auton Robot* 19, 41-52 (2005).
5. Brooks, C.A., Iagnemma, K.: Visual Detection of Novel Terrain via Two-Class Classification. In: *Proceedings of the 2009 ACM Symposium on Applied Computing*, pp. 1145-1150 (2009).
6. Pereira, G.A.S., Pimenta, L.C.A., Chaimowicz, L., Fonseca, A.F., de Almeida, D.S.C., Correa, L.Q., Mesquita, R.C., Campos, F.M.: Robot Navigation in Multi-Terrain Outdoor Environments. *Int J Robot Res* 28, 685-700 (2009).
7. Pietikäinen, M., Nurmela, T., Mäenpää, T., Turtinen, M.: View-Based Recognition of Real-World Textures. *Pattern Recogn* 37, 313-323 (2004).
8. Liao, S., Chung A.C.S.: Texture Classification by Using Advanced Local Binary Patterns and Spatial Distribution of Dominant Patterns. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1221-1224 (2007).
9. Columbia Utrecht Reflectance and Texture Database, <http://ww1.cs.columbia.edu/CAVE//software/curet/>
10. Turk, M., Pentland, A.: Eigenfaces for Recognition, *J Cognitive Neurosci* 3, 71-86 (1991).
11. Kahraman, F., Stegmann, M.B.: Towards Illumination-Invariant Localization of Faces Using Active Appearance Models. In: *7th Nordic Signal Processing Symposium*, pp. 4 (2006).
12. ROBOTIS Co., LTD, <http://www.robotis.com>
13. Tsai, D.M., Tseng, C.F.: Surface Roughness Classification for Castings. *Pattern Recogn* 32, 389-405 (1999).
14. Lambert, A., Gruyer, D., Pierre, G.S., Ndjeng, A.N.: Collision Probability Assessment for Speed Control. In: *11th International IEEE Conference on Intelligent Transportation Systems*, pp. 1043-1048 (2008).
15. Selekwa, M.F., Dunlap, D.D., Shi, D., Collins, E.G.: Robot Navigation in Very Cluttered Environments by Preference-Based Fuzzy Behaviors. *Robot Auton Syst* 56, 231-246 (2008).
16. Ward, K., Zelinsky, A.: Acquiring Mobile Robot Behaviors by Learning Trajectory Velocities. *Auton Robot* 9, 113-133 (2000).