Autonomous Terrain Characterisation and Modelling for Dynamic Control of Unmanned Vehicles

A. Talukder, R. Manduchi*, R. Castano, K. Owens, L. Matthies, A. Castano, R. Hogg

Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA. Tel. (818)354-1000 – Fax (818)393-3302
Email: [ashit.talukder, rebecca.castano, larry.h.matthies, andres.castano, robert.w.hogg]@.jpl.nasa.gov

**University of California at Santa Cruz*
Santa Cruz, CA. Tel. (831)459-1479 – Fax (818)459-4829. Email: manduchi@soe.ucsc.edu

Abstract

An often-ignored aspect of unmanned cross-country vehicles is the dynamic response of the vehicle on different terrain. We discuss techniques to predict the dynamic vehicle response to various natural obstacles. This method can then be used to adjust the vehicle dynamics to optimize performance (e.g. speed) while ensuring that the vehicle is not damaged. This capability opens up a new area of obstacle negotiation for UGVs, where the vehicle moves over certain obstacles, rather than avoiding them, thereby resulting in more effective achievement of objectives. Robust obstacle negotiation and vehicle dynamics prediction requires several key technologies that will be discussed in this paper. We detect and segment (label) obstacles using a novel 3D obstacle algorithm. The material of each labelled obstacle (rock, vegetation, etc.) is then determined using a texture or color classification scheme. Terrain load-bearing surface models are then constructed using vertical springs to model the compressibility and traversability of each obstacle in front of the vehicle. The terrain model is then combined with the vehicle suspension model to yield an estimate of the maximum safe velocity, and predict the vehicle dynamics as the vehicle follows a path. This end-to-end obstacle negotiation system is envisioned to be useful in optimized path planning and vehicle navigation in terrain conditions cluttered with vegetation, bushes, rocks, etc. Results on natural terrain with various natural materials are presented.

Keywords: Classification, geometrical reasoning, navigation, obstacle detection obstacle negotiation, terrain perception, terrain modeling, vehicle modeling.

1. Introduction

Driving in cross-country vegetated environments requires a higher level of scene understanding and vehicle control than in arid terrain or static urban environments. In the latter cases the trafficability of a given path is determined solely by the presence or the absence of obstacles that may hamper the vehicle's moving. Obstacles should be avoided when they are tall enough to harm the vehicle (such as a wall, a telephone pole or a big rock), or, if they are of small dimension, they may be driven over at low speed. The underlying assumption in these kinds of environments is that all materials in the scene are uncompressible. Thus, the "load-bearing surface" along a path corresponds to the actual visible surface, meaning that geometry description (as acquired by range sensors such as stereo cameras or lidars) provides enough perceptual information to control the vehicle, that is, to decide the most appropriate path and optimal velocity.

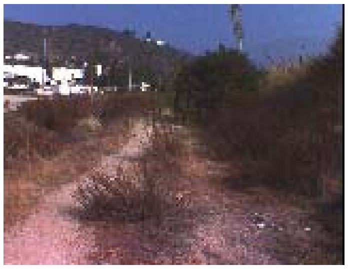


***Figure 1**: Example of natural terrain that requires obstacle negotiation*

In contrast, objects or surfaces composed by vegetative material are often rather compressible. Think for example of a thin bush (see Figure 1) or a tuft of tall grass: such "obstacles" are indeed traversable by a suitably sized vehicle, even when their height is such that they would harm the vehicle were they made of "hard" material (such as rock, concrete or wood.) A system unaware of the compressibility properties of such materials would avoid all such "obstacles", resulting in unnecessary tortuous and inefficient paths. An extreme case is given by a field of tall grass, where the vehicle is faced with a continuum of apparent obstacles. Thus, traditional sensing and control mechanisms should be revisited for autonomous navigation in vegetated terrain, where obstacle negotiation is an important and complex component of the problem.

This paper presents a combined approach to obstacle negotiation, comprising the operations of obstacle detection, terrain cover classification and compressibility characterization, and dynamic vehicle modeling (see Figure 2). Each module in Figure 2 provides information that is used for the characterization of vehicle dynamics over the terrain. The obstacle detector provides 3D shape/geometrical information about each obstacle. Texture and color reasoning combined with the 3D geometrical information provide information about the material class of each obstacle. This shape and material information is then used by the vehicle velocity control module (green block in Figure 2).

We introduce a simple spring/damper model for representing the reactive characteristics of an obstacle. If the obstacle is made of uncompressible material (e.g., a rock), then the spring constant is infinite; otherwise, it is assigned a finite value depending on the material. The vertical acceleration of the vehicle as it traverses a candidate path at a given velocity is predicted by considering the height profile of the path, the dynamic characteristics of the vehicle (modeled by a spring/damper parallel), and the reactive characteristics of obstacles along the path. This predicted acceleration profile is at the basis of our obstacle negotiation strategy, allowing us to determine the "optimal" velocity for the traversal, that is, the maximum velocity such that the peak vertical acceleration is below a preassigned value.

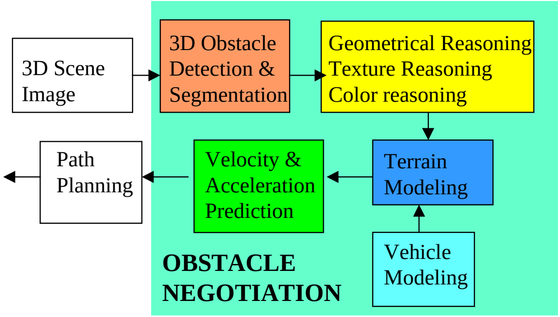

3D Scene Image
3D Obstacle Detection & Segmentation
Geometrical Reasoning Texture Reasoning Color reasoning
Terrain Modeling
Velocity & Acceleration Prediction
Path Planning
Vehicle Modeling
OBSTACLE NEGOTIATION

***Figure 2**: The modules in our terrain negotiation procedure. Shaded/color portions show the modules discussed in this paper for obstacle negotiation.*

The paper is organized as follows: Our obstacle detection and segmentation algorithm is discussed in Section 2. Section 3 details our obstacle reasoning algorithms, including geometrical shape-based reasoning (Section 3.1), terrain material classification using texture (Section 3.2) and color classifiers (Section 3.3). In Section 4, we discuss our loadbearing surface modelling technique for prediction of vehicle dynamics as it negotiates detected obstacles. Results are presented in Section 5.

2. Obstacle characterisation algorithm

2.1. Obstacle detection

Obstacle detection is one of the main requirements of mobile robots. Positive obstacles correspond to objects that project upwards from the ground, such as bushes, trees, grass, poles, road signs, etc. Detection of positive obstacles have previously involved fitting of ground planes to 3D elevation maps, and classifying all groups of pixels above the ground plane as obstacles, or using slope measures along columns in the range image [Bellutta00] to locate obstacles, followed by 2D blob colouring to remove small obstacles.

The plane fitting technique fails when the scene has elevation variations, which occurs frequently in natural terrain. The column-wise scanning technique works well when obstacles are vertically oriented, but fails for slanted sloped objects.

We develop a true 3D obstacle detector that searches along cones at each location in 3D space Note that a truncated cone centered about a location projects onto a truncated triangle in the image plane. The height of each projected triangle is proportional to cone height and distance of the point from the image plane. Details of our obstacle detector are provided in [Talukder02].

2.2. Obstacle segmentation

Obstacle segmentation for mobile robotics have mostly used 2D blob-based measures to reduce false obstacle detection [Bellutta00]. This is inadequate when obstacles that are separated in depth in 3D space are adjacent in 2D image space due to overlap in their x,y coordinates.

Our 3D obstacle detection algorithm using searches along cones/triangles inherently does obstacle segmentation as a by-product. This occurs by coloring *valid* obstacle pixels in the same cone (triangle) to one label as the algorithm progresses in row-scan order. All these valid obstacle pixels in the cone are recolored if any of them are already labeled (based on searches along previous pixels in the row-scan order). This results in a truly 3D segmented obstacle image, where obstacles that are spatially adjacent in x,y (2D image space) are assigned different labels if they are far away in depth. The obstacle segmentation algorithm is detailed in [Talukder02].

3. Obstacle Reasoning

Obstacle material reasoning provide useful information for the terrain modelling and path planning algorithms. Robust obstacle reasoning should be able to (1) prune false obstacles, and (2) provide adequate information to model the response of each obstacle to vehicle dynamics. We use three disparate obstacle reasoning methods, each of which provides unique (and complementary) information about each obstacle. We use geometrical 3D obstacle information from the obstacle detector, and material classification using texture and color information. We then fuse the information from these modules that yields more robust obstacle pruning and material reasoning results than each individual method.

3.1. Geometrical and Shape-based Reasoning

Prior 3-D model-based reasoning for robotics [Hoover98] typically require converting the 3-D point data into a mesh representation, which is a complex operation. We compute 3-D geometrical features from the raw point-cloud data, which enables real-time analysis.

We extract simple 3-D geometrical measures from each obstacle, including the 3D perimeter, and average/maximum obstacle slope & relative height. *These geometrical measures are automatically derived during the obstacle segmentation process, without any extra computational overhead*. If any of the five variables have a value less than the pre-selected thresholds (i.e. object is too small, low average.maximum slope, etc.), it is rejected as a false obstacle. Figure 5a shows examples of false-objects (flat areas) that are correctly rejected (red coloured areas) after obstacle reasoning.

3.2. Texture-based Material Classification

Visual texture can provide valuable information about the identity of imaged objects. Obstacles within a class, such as bushes, often have similar texture signatures and thus, when combined with obstacle detection, texture can be used to discriminate among several classes of obstacles. Texture can perform well in cases where color classification is ineffective or not possible. For example, it works well on infrared images taken at night. Furthermore, there are classes of materials such as dry vegetation, bark, and soil that are difficult to distinguish with color but can be readily distinguished using texture.

Texture measures the local intensity variation at different orientations and spatial frequencies. Local texture features condense information from a small neighborhood of a given pixel. Our features are obtained by convolving each filter in a multiscale, multioriented Gabor filter bank with the original image. Such feature operators have proven to be effective and have become a standard choice in most of the recent texture analysis algorithms in the literature. With the extracted features, we use a classifier to label the image pixels. The classifier models the probability distribution function of the texture features for each obstacle class as a mixture of three Gaussians, and performs a Maximum Likelihood (ML) classification. The Expectation-Maximization (EM) algorithm is used to train each class of the classifier. The classification method is described in detail in [Castano 2001].

3.3. Color-based Material Classification

Color features are attractive because they provide useful information about terrain type and produce classification results with small computational cost. We used the Bayesian color classifier that we developed for the DEMO III project [Shoemaker98], which uses Gaussian mixtures to model the classconditional color likelihoods.

Our system uses three terrain surface classes: "green vegetation", "dry vegetation", and "soilrock". A fourth default class ("outlier") accounts for surfaces whose apparent color is not well represented by any of such three classes. Green vegetation and dry vegetation are typically well separable in color space. One reason to keep these two classes separated (rather than having a common "vegetation" class) is that the dry vegetation class includes bark and therefore tree trunks, which are usually not traversable. Green vegetation, instead, typically includes only grass, bushes and leaves. Figure 3 shows color classification results on natural terrain. Note that in some instances it may be very difficult to separate bark from other kinds of dry vegetation, or dry vegetation from some types of soil, based solely on color [Roberts93]. Discrimination can be improved in these cases by fusing color information with evidence from other visual features (such as texture and shape analysis.) The measured radiance spectrum is a function of the illuminant spectrum and surface reflectance. To correct for changes in the sunlight spectrum, standard white point calibration procedures (often implemented in hardware onboard the camera) may be used. Note however that it is not possible in principle to correct for both sunlight and diffuse ambient light at the same time, due to their different spectral characteristics. Our current approach to the problem is to correct for the sunlight spectrum using a white reference, and to deal with other variations of the illuminant using a rich enough statistical model trained with a large and possibly diversified set of training data.

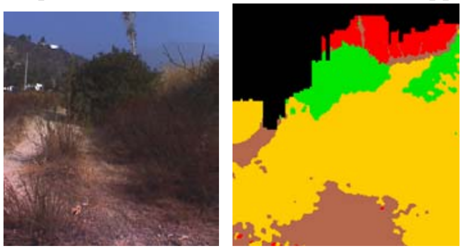


Figure 3*. Original & Color-based classification (from [Bellutta00] (brown = soil; yellow = dry vegetation; green = green vegetation; red = outlier). Pixels further than 50 m have not been classified*

4. Terrain Load Bearing Surface Modeling for Vehicle Dynamics

One important desired feature in UGVs is the ability to know which obstacles to avoid, the intelligence to slow down when hard “uncompressible” obstacles are detected and continue driving normally when traversable (“compressible”) objects such as tall grass or thin bushes are encountered. The modules discussed so far provide information about the location, shape and size of each obstacle, and the material or class of each obstacle (stone, vegetation, etc.) This information can be used to estimate the “compressibility” of each object in the scene. We use springs to model terrain compressibility of each obstacle. The vehicle dynamic suspension is also modelled using another spring, which is based on work done previously at JPL [Rankin98]. Our work is an improvement to that suggested in [Rankin98] in that it con-

[1] Outliers are detected by thresholding the likelihood of the color of each pixel according to our model [Ripley96].

siders terrain compressibility, a critical issue in obstacle negotiation.

Therefore, as shown in Figure 4, we view the vehicle and its load-bearing surface as a mass-spring system, where the quarter-model of the vehicle (one wheel in a four-wheeled vehicle) and terrain are each modelled as springs, each with its own spring parameters. We assume that the spring constant K_O of each terrain obstacle class is unique and known a-priori (our initial tests indicate the feasibility of using learning to improve the spring models online, but is not discussed here due to space constraints). K_O is a function of the obstacle material (described mathematically by the material's shear/elastic modulus), ***and*** the height of the obstacle. Stone/rocks and logs have large shear/elastic modulus), tall grass has low shear/elastic modulus, and bushes have a medium modulus. For a given obstacle of known material (and known shear/elastic modulus), we assume that the spring constant is proportional to its height, similar to spring coil models, i.e. the taller an obstacle, the larger is its spring constant K_O. *Note that the obstacle height is obtained automatically from our obstacle segmentation routine*, and therefore a separate ground-plane fitting algorithm is not required for height estimation.

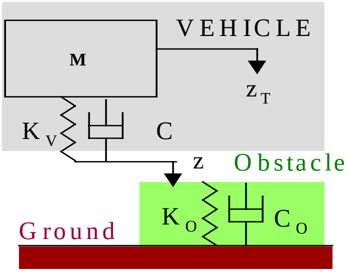


***Figure 4**: Terrain-vehicle spring model system for vehicle velocity control*

In order to limit the forces that the vehicle is subjected to while driving over terrain, we derive a relation between the vehicle's velocity and the vertical acceleration. The model shown in Fig. 4 can be used to approximate the dynamic motion of a quarter vehicle suspension system over various terrain types, where $M, K, C, z,$ and z_T are the quarter vehicle mass, the ***effective*** suspension spring constant of the vehicle and terrain class in front, the ***effective*** suspension damping, and the displacements from equilibrium of the mass M and tire axle. The equation describing the motion of this system is well known

$$M\frac{d^2z}{dt^2} = -K(z - z_T) - C\frac{d}{dt}(z - z_T) \qquad (1)$$

The effective spring constant K of two springs in series (vehicle and obstacle) with each other are equivalent to two resistors in parallel, i.e. $1/K = 1/K_V + 1/K_O$. Therefore, the presence of a vegetation obstacle would lower the effective spring constant, and stone obstacles would result in larger effective spring constants. For current purposes, we assume that the spring damping coefficient is the same for all objects and the vehicle suspension.

For this effective spring-mass system, we now derive our velocity and acceleration prediction techniques for a case where only one obstacle of a known material type is present in front of the vehicle. *We shall later generalize this to the case where several obstacles of different material types are present in the scene*. It is clear from equation 1 that if the relative displacement and its derivative are known, the vertical acceleration of the vehicle can be computed. For a given vehicle and a single obstacle with known material class, equation (1) simplifies to

$$\frac{d^2r}{dt^2} + \frac{C}{M}\frac{dr}{dt} + \frac{K}{M}r = \frac{d^2h}{dt^2}$$

where the relative displacement, r, and the road height, h, are given by $r = z - z_T, h = -z_T$.

We use exactly the approach used in [Rankin98] to solve for the dynamic control problem, for a given obstacle material type (and therefore a known effective spring constant for the vehicle-obstacle spring system). Only the outline of the derivation is provided, and the reader is referred to [Rankin98] for details. Using $x_1 = r, x_2 = dr/dt$, in matrix notation, $\dot{x} + Ax = \begin{pmatrix} 0 \\ \frac{d^2h}{dt^2} \end{pmatrix}$ where $A = \begin{pmatrix} 0 & -1 \\ \frac{K}{M} & \frac{C}{M} \end{pmatrix}, x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, yields the solution

$$x(t) = e^{-At}x_0 + \int_0^t e^{A(\tau - t)} \begin{pmatrix} 0 \\ \frac{d^2h}{dt^2} \end{pmatrix} d\tau \qquad (2)$$

Ignoring initial conditions for now (we will discuss them later), we change the dependent variable to distance (since the data is available as a range/elevation map), resulting in $x(l) = U\int_0^l e^{\frac{A}{U}(s-l)} \begin{pmatrix} 0 \\ \frac{d^2H}{ds^2} \end{pmatrix} ds$ where $h(t) = H(s(t))$. After simplification [Rankin98], we can write the vertical vehicle acceleration as,

$$a = c_2 f_2^n - Uc_1 f_1^n - \frac{\Delta l_n}{U} c_3 f_3^n \qquad (3),$$

with $c_1 = \frac{C}{M}, c_2 = \frac{K}{M} - (\frac{C}{M})^2, \text{and} c_3 = \frac{K}{M}\frac{C}{M} - (\frac{C}{M})^3$, and f_1, f_2, and f_3 are functions of the terrain elevation slope and its double derivative.

To avoid damaging the vehicle, we ensure that the vertical acceleration is bounded and choose U such that

$$|a| = |c_2 f_2^n - Uc_1 f_1^n - \frac{\Delta l_n}{U} c_3 f_3^n| \leq a_{MAX} \qquad (4),$$

which would be the maximum-valued root of

$$U^2 c_1 f_1^{\max} + U(|c_2| f_2^{\max} - a_{\max}) + \Delta l_{\max} c_3 f_3^{\max} = 0 \qquad (5).$$

For a single obstacle in front of the vehicle, Eq. (3) is used to predict the acceleration of the vehicle when the material class, and therefore the spring model parameters of the obstacle is known, and Eq. (4) yields the maximum safe velocity when the vehicle traverses the obstacle. In practice, we model the elevation map along the path followed by the vehicle as cubic-splines to accommodate missing data points in the range/elevation map directly in front of the vehicle.

For the case when several obstacles are present in front of the vehicle, we divide the traversal path into segments, corresponding to non-obstacle segments, and obstacle segments. For N obstacles in front the vehicle, the traversal path is therefore divided into (2N + 1) segments. We apply the velocity control and acceleration prediction algorithms on each segment that has its own spring model parameters; for example, the effective spring constant for ground segments is $K = K_V$ (since K_O = infinity).

An issue that arises in modeling the system dynamics are the initial conditions of the system $x_0 = [r \;\; dr/dt]$. When the vehicle starts from a stationary state, the initial conditions (Eq. 2) can be assumed to be zero. For a case where N obstacles are present, we assume that the initial conditions at the beginning of each of the (2N + 1) segments is $x_0 = [0 \;\; 0]$. This is a reasonable assumption for the n'th segment of the vehicle's path, ***if*** the vehicle zero vertical velocity at the end of the (n-1)'th segment. This holds true if the terrain at the *end of the (n-1)'th segment* has relatively low elevation changes. This is a realistic assumption since the path is segmented out based on the presence of obstacles which are generally characterized by sharp changes in elevation.

Incorporating the true initial conditions at the beginning of each of the (2N + 1) segments is, however, expected to give more accurate predictions of vehicle dynamics and will be implemented in future work.

5. Results

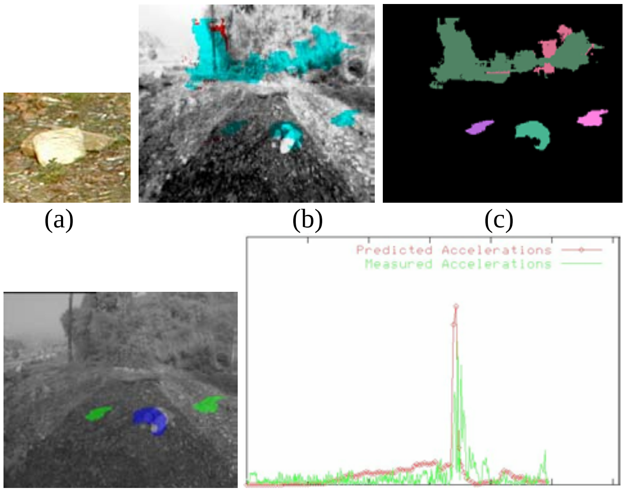
We present results of our end-to-end obstacle negotiation algorithms. This includes the obstacle detector, and obstacle labeling (Section 2), obstacle reasoning using shape and texture classifiers (Section 3), and velocity control and terrain modelling (Section 4).

We tested our terrain/obstacle negotiation algorithms on terrain comprised of 4 traversible objects, namely two logs, a bush, and a stone. Grayscale stereo cameras were used, due to which material classification was done using the texture classifier (Section 3.2) only. Range data was obtained using JPL stereo algorithms. An inertial measurement unit (IMU) was placed on the vehicle to measure vehicle velocity and accelerations. These IMU measurements were used to verify the accuracy of our terrain modeling & velocity control prediction algorithms.

We used three obstacle material classes for the texture classifier: log, bush and rock. We trained on several images of the obstacles, using only image pixels of the obstacle, and tested the classifier on the obstacles detected in complete sequences of images. During testing, we did not employ an outlier detector, thus each pixel is assigned to one of the three obstacle classes. The classifier is very effective for the obstacles for which it was trained, however the assigned label is only meaningful for pixels representing obstacles. Regions in the image representing classes not used for training, such as sky or flat ground, receive arbitrary class labels. In each test image, we computed a class label for every obstacle pixel.

We consider combined texture/color classification with obstacle segmentation information to achieve better obstacle reasoning. Prior work on Demo III [Bellutta00] used a majority-based decision using the color classifier on each 2D obstacle blob. In our preliminary tests, we classify each segmented 3D obstacle using a majority classification voting. Using true 3D segmented obstacles is expected to yield better material classification than texture classification on 2D obstacle blobs that may actually contain 2 or more overlapping obstacles.

Figure 5a shows a color image of a stone in front of the vehicle, and Figure 5b shows the true obstacles (blue) and the rejected false obstacles (red) after shape/geometrical reasoning. Figure 5c shows the segmented obstacle map with each obstacle assigned a unique color. The texture classification results on obstacles within 15 metres from the vehicle are shown in Fig. 5b, with stone regions as blue and vegetated obstacles as green. Note that the fusion of the texture material classifier and the obstacle detector results in more robust reasoning. The stone is classified as a valid obstacle since it is of sufficiently height (based on geometrical reasoning) and is classified as rock by the texture classifier. The other two regions (green patches in Figure 6b) that are



(a) (b) (c)

(d) (e)

***Figure 5**: (a) Stone obstacle (b)detected obstacles (blue) and rejected (red) (c) Segmented obstacle image (d) texture classified image of obstacles within 15 metres (blue- stone, green- vegetation) and (e) Predicted vs. actual acceleration for stone obstacle.*

incorrectly labeled as obstacles by the shape reasoning procedure are correctly classified as vegetation (low grass) by the texture classifier. Using a majority-based decision on labeled obstacles, rather than pixel-based classification, makes the texture material classification performance significantly better than individual pixel-based classification. The absolute value of the predicted acceleration for the vehicle as it moves at a constant speed is shown as a red line in Fig. 5d. Note that the absolute acceleration value is required to assess vehicle tolerance limits (the sign, i.e. direction of acceleration is not needed). The absolute value of the true acceleration of the vehicle measured from the IMU is shown as a green plot. The sharp peak corresponds to the time that the vehicle moves over the stone.

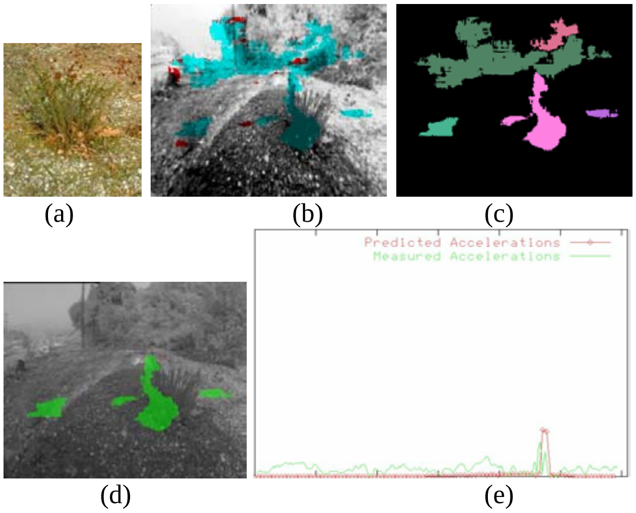


Figure 6*: (a) Bush obstacle (b)detected obstacles (blue) and rejected (red) (c) Segmented obstacle image (d) texture classified image of obstacles within 15 metres (green-vegetation) and (e) Predicted vs. actual acceleration for bush obstacle*

Figure 6 shows the results as the vehicle negotiates a medium-sized bush. The obstacle detection algorithm correctly detects (Figure 6b) and segments (Figure 6c) the bush in the foreground and the background obstacles. The texture classifier classifies the bush as a vegetation class (green), which results in a low effective spring contant for that segment of the terrain. Therefore, even though the bush is actually significantly taller than the stone in Figure 6, the velocity control algorithm predicts a small acceleration peak on the bush (red line plot in Figure 6e), which is verified via the true vehicle dynamics IMU measurements (green line plot in Figure 6e).

6. Conclusions and Future Work

We have presented an integrated approach to obstacle negotiation in sparsely vegetated terrain. This technique represents a clear improvement with respect to traditional obstacle avoidance procedures, and has the potential to dramatically increase the efficiency of autonomous vehicles in cross-country environments. While our preliminary experiments show very promising results, several hurdles lay ahead and are the object of current research. First, terrain cover classification based on color and texture needs to be made more robust to various environmental conditions and more computationally efficient. We are also considering different sensors, such as multispectral cameras in the thermal infrared domain and lidars, to achieve terrain classification when color or texture analysis fail. Second, pixel-wise classification, as obtained by color or texture, should be combined with explicit shape reasoning in order to correctly characterize an obstacle. For example, an obstacle shaped like a thin pole may correspond to a tree trunk and should not be confused with a dry bush, even if the color may be similar in the two cases. Third, the relationship between material type/ obstacle size and spring constant in our compressibility model needs to be established through thorough testing. In particular, we are exploring the possibility of online learning, where spring constants are refined by comparing predicted and post-facto measured acceleration profiles.

Acknowledgements

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the DARPA-ITO Mobile Autonomous Robot Software (MARS) Program through an agreement with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

7. References

[Bellutta00] P. Bellutta, R. Manduchi, L Matthies, K. Owens, A. Rankin, “Terrain Perception for DEMO III”, *2000 Intelligent Vehicles Conference*,, 2000.

[Castano01] Castano, R., R. Manduchi, J. Fox, “Classification Experiments on Real-World Textures,” Empirical Evaluation Methods in Computer Vision, Kuaii, HI, Dec. 2001.

[Hoover98] Hoover, et. al. “The space envelope: a representation for 3D scenes”, *Computer Vision and Image Understanding*, vol.69, no.3, 310-29, March 1998.

[Roberts93] D. Roberts, M. Smith, J. Adams, “Green Vegetation, Nonphotosynthetic Vegetation, and Soils in AVIRIS Data”, Remote Sens. Environ., 44:255-269, 1993.

[Rankin98] A. Rankin, K. Owens, L. Matthies, T. Litwin, “Terrain-Adaptive Gaze and Velocity Control for UGV Obstacle Detection”, Association for Unmanned Vehicle Systems International Annual Symposium , June 8-12, 1998.

[Shoemaker98] C.M. Shoemaker, J.A. Bornstein, “The Demo III UGV Program: A Testbed for Autonomous Navigation Research”, Proceedings of the IEEE International Symposium on Intelligent Control, Gaithersburg, MD, September 1998.

[Talukder02] A. Talukder, R. Manduchi, L. Matthies, A. Rankin, “Fast and Reliable Obstacle Detection and Segmentation for cross country navigation”, IEEE IV 2002, June, 2002.