Autonomous Robots 18, 81–102, 2005

Obstacle Detection and Terrain Classification for Autonomous Off-Road Navigation

R. MANDUCHI
University of California at Santa Cruz, Santa Cruz, CA 95064, USA
manduchi@soe.uesc.edu

A. CASTANO, A. TALUKDER AND L. MATTHIES
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA

Abstract. Autonomous navigation in cross-country environments presents many new challenges with respect to more traditional, urban environments. The lack of highly structured components in the scene complicates the design of even basic functionalities such as obstacle detection. In addition to the geometric description of the scene, terrain typing is also an important component of the perceptual system. Recognizing the different classes of terrain and obstacles enables the path planner to choose the most efficient route toward the desired goal.

This paper presents new sensor processing algorithms that are suitable for cross-country autonomous navigation. We consider two sensor systems that complement each other in an ideal sensor suite: a color stereo camera, and a single axis ladar. We propose an obstacle detection technique, based on stereo range measurements, that does not rely on typical structural assumption on the scene (such as the presence of a visible ground plane); a color-based classification system to label the detected obstacles according to a set of terrain classes; and an algorithm for the analysis of ladar data that allows one to discriminate between grass and obstacles (such as tree trunks or rocks), even when such obstacles are partially hidden in the grass. These algorithms have been developed and implemented by the Jet Propulsion Laboratory (JPL) as part of its involvement in a number of projects sponsored by the US Department of Defense, and have enabled safe autonomous navigation in high-vegetated, off-road terrain.

Keywords: obstacle detection, terrain classification, color classification, ladar classification, autonomous navigation

1. Introduction

Robots that can drive autonomously in off–road environments have received a good deal of attention in recent years. The US Department of Defense through its various agencies has been the major sponsor of research in this field. Notable examples include the DEMO I, II and III projects (Shoemaker and Bornstein, 1998) and the DARPA Grand Challenge (http://www.grandchallenge.org). Indeed, in the 2001 Defense Authorization Act, the U.S. Congress set a goal that by 2015 one third of the operational ground vehicle be unmanned (http://www.gao.gov/new.items/d01311.pdf). Autonomous off-road robots will be employed not only in military operations, but also in civilian applications such as wide-area environment monitoring, disaster recovering, search-and-rescue activities, as well as planetary exploration.

Possibly the single biggest technological challenge for these systems is the ability to sense the environment and to use such perceptual information for control. Indeed, even if a robot is equipped with a Global Positioning System (GPS) and an Inertial Navigation Unit (IMU), reliable environment sensing is necessary for autonomous operation beyond the line of sight from the operator. Lacking perception capabilities, the robot would have to rely solely on self-localization (for

example, using GPS) and on prior environment maps. Unfortunately, the resolution of GPS is too low for tasks such as obstacle avoidance. Furthermore, maps normally have too low a resolution for small obstacles to be included, lack accurate elevation information, and become obsolete. Thus, environment sensing is essential for the task of efficient navigation over long distances.

It should be clear that driving in outdoor, non-urban environments is more challenging than driving indoors or in urban scenarios (as highlighted by the 2004 DARPA Grand Challenge (http://www.grandchallenge.org), whereby all competing vehicles failed within the first 7 miles of the 142 miles route). In the latter case, the environment is highly structured, which typically translates into simpler sensing and action strategies. For example, in urban situations one may expect the ground surface in front of the robot be planar, which helps detecting obstacles (anything sticking out of the ground plane should be avoided). In contrast, on a bumpy dirt road the robot should constantly determine which bumps and holes are small enough to be negotiated (possibly by slowing down) and which ones should be avoided. Vegetated terrain introduces one more degree of freedom to the problem: what is considered an "obstacle" from a purely geometric point of view, may not represent a danger for the vehicle if it is composed of compressible vegetation (for example, a tuft of tall grass or a small bush). Other challenging situations include the presence of negative obstacles (such as ditches), elements such as water, mud or snow, and adverse atmospheric conditions such as fog.

This paper addresses a set of perception tasks that are at the core of any control system for efficient autonomous navigation in cross-country environments. More precisely, we introduce new algorithms for (1) obstacle detection and (2) terrain cover classification. The problem of obstacle detection and avoidance is well studied in robotics, however, existing algorithms apply mostly to urban or indoor environments and don't work well in off-road conditions. This is because typical assumptions about the scene, such as the existence of flat ground surface, do not hold in this case. Our new algorithm analyzes the slant of surface patches in front of the vehicle, and identifies patches that are steep enough to represent a hurdle for the vehicle. The analysis is carried out on the range data produced by a laser rangefinder (ladar) or, as in the examples in this paper, by a stereo system. Besides detecting visible obstacle points, our algorithm clusters such points into distinct surface segments, which helps building a symbolic representation of the scene in terms of an obstacle map.

While geometric scene description and obstacle detection are essential tasks, they are not sufficient to ensure efficient navigation on cross-country terrain. In these environments, there are several possible terrain types (such as soil, grass, mud) that should be negotiated by the vehicle in different ways. An effective description of the outside world should consist of the combination of geometric *and* terrain type information, coupled with adaptive control strategies (Bellutta et al., 2000; Hebert et al., 2002). The DEMO III project (Shoemaker and Bornstein, 1998) pioneered this concept, resulting in possibly the most advanced autonomous vehicle for cross-country navigation. We present two approaches to terrain cover perception, one based on stereo and color analysis (from color cameras), and the other based on range data processing (from a ladar). These two systems are somewhat complementary in the cues they use to discriminate between different terrain cover classes: surface reflectivity in the first case, local range statistics in the second one. Color classification allows one to recognize a certain number of distinctive classes such as grass and foliage, dry vegetation and bark, soil and rocks. Since stereo data is co-registered with color data, it is possible to label the obstacle detected on the basis of range information with the terrain type estimated by color analysis. The main challenges of color-based classification are the inherent ambiguity of the reflectivity spectrum for some classes of interest (such as dry grass and soil), the effect of the illumination spectrum on the perceived color (the so-called *color constancy* problem), and chromatic shift due to atmospheric effects. The analysis of local range statistics from ladar data, which is at the core of our second technique, can be used to successfully discriminate grass or foliage from other smooth surfaces such as rocks or tree trunks. This "range texture" approach is very promising, allowing one to detect surfaces even when they are partially hidden by grass.

The two systems for terrain typing (color-stereo and ladar) discussed in this paper have complementary functionalities in the context of off-road autonomous navigation, and ideally should both be part of the sensor suite of a robot for such application, as shown in 1. A color stereo camera is most effective for detecting and characterizing isolated obstacles, as well as for capturing elevation profiles of the scene. A single-axis ladar placed in the lower portion of the front of the

vehicle could be used for safe navigation in tall grass, where obstacles may be partially hidden by vegetation and therefore may not be detected by stereo analysis. Together, these two systems may enable a new level of autonomy in a wide variety of environmental conditions.

The algorithms presented in this paper have been developed and implemented by the Jet Propulsion Laboratory (JPL) as part of its involvement in a number of projects (ARL Demo III, DARPA Mobile Autonomous Robot Software (MARS), DARPA Tactical Mobile Robotics (TMR), and DARPA Perceptor).

This article is organized as follows. Section 2 presents our geometry-based obstacle detection and segmentation algorithm. Section 3 describes our two strategies for terrain cover classification: color analysis (Section 3.1) and range texture analysis (Section 3.2). Section 4 has the conclusions.

2. Obstacle Detection

Path planning for autonomous vehicles requires that the map of all potential obstacles be produced in real time using available sensor data. The obstacle-free candidate paths leading to the desired position may then be compared in terms of their hindrance (measured, for example, by the amount of steering involved (Lacaze et al., 1998; Badal et al., 1994).

For navigation indoors or in structured environments (such as roads), obstacles are simply defined as surface elements that are higher then the ground plane by some amount. Thus, assuming that elevation information is available (by means of a stereo camera or a ladar), obstacle detection (OD) algorithms normally rely on the "flat world" assumption (Zhang et al., 1994; Williamson and Thorpe, 1998; Baten et al., 1998; Broggi et al., 2000; Badal et al., 1994; Singh and Keller, 1991)). While this assumption works well for urban/indoor situations, it does not apply to off-road, cross-country environments, where the geometry of the terrain can hardly be modeled as a planar surface (Batavia and Singh, 2002; Hong et al., 2000), see also Fig. 7). If the geometry of the terrain in front of the vehicle is available (from stereo or ladar), it is still possible to assess whether a candidate path can be traversed without damage to the robot. In principle, one could determine the traversability of a given path by simulating the placement of a 3-D vehicle model over the computed elevation map, and verifying that all wheels are touching the ground while leaving the bottom of the vehicle clear (Lacaze et al., 2002). This procedure, however, besides being computationally heavy, requires a high-resolution elevation map.

Indeed, solutions for obstacle detection that use elevation maps (Lacaze et al., 2002) or voxel-based representations (Hebert et al., 2002) of the world, require a change in the input data structure, which introduce additional data transformation steps prior to visual obstacle detection. This could produce sampling artifacts at object boundaries, and requires substantial computational power and storage space. In contrast to these approaches, our OD algorithm works directly with measured 3-D point clouds. Obstacles are detected by measuring slope and height of visible surface patches, an operation that can be carried out directly in the range image domain. An earlier version of our algorithm (Matthies and Grandjean, 1994; Matthies et al., 1996, 1998; Bellutta et al., 2000) was based on the

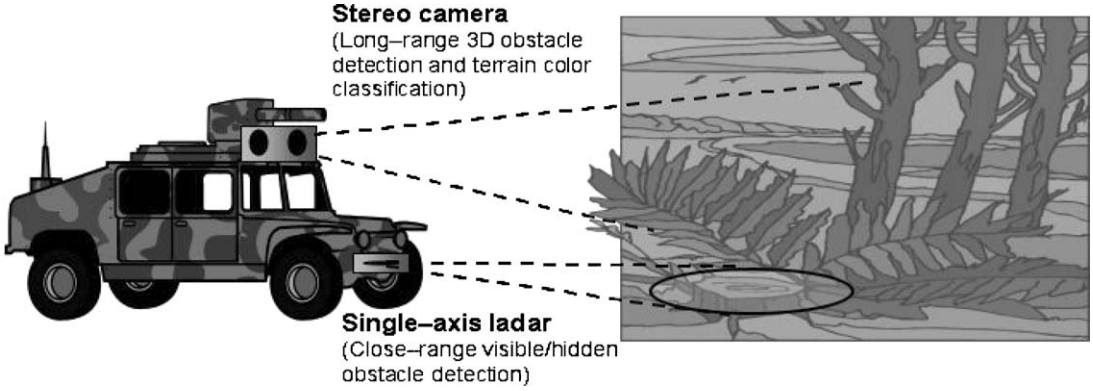

Figure 1. Concept figure for our considered sensor suite.

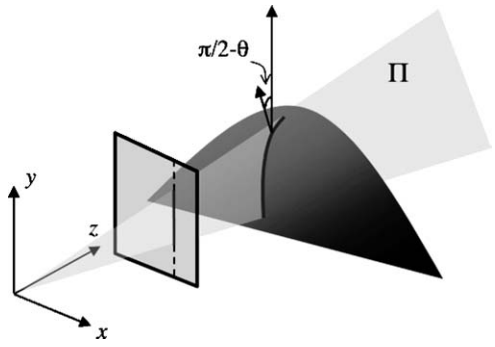

Figure 2. The viewing geometry for the column-based obstacle detector. q is the slope angle. The origin of the axes corresponds with the cameras focal point.

analysis of 1-D profiles corresponding to range values in each pixel column in the image plane.[1] These measurements sample the trace left by the visible surface on the slicing planes Π defined by each pixel column and the focal point of the camera (see Fig. 2). Assume that a world reference frame has been defined, with x and y axes parallel to the image plane, origin coincident with the focal point, and y axis pointing vertically. Assume further that the y axis is parallel to the image columns (this can be obtained by rectifying the image after camera calibration). One may then use the slope of the 1-D range profile with respect to the horizontal plane to reason about obstacles. For example, Fig. 2 shows an instance of a 1-D range profile, where slant (θ) and height (H) are shown for two different surface patches. Obstacles correspond to ramps with a slope above a certain threshold and spanning a minimum height. The rationale behind this approach is simple: if a surface patch has limited slope, we may assume that it belongs to the ground surface (for example, as part of a path leading to a plateau), and therefore it is traversable. If a patch is steep but of small size, it corresponds to an obstacle short enough to be negotiable by the vehicle. For example, the lower patch in 2 may be considered traversable, while the higher patch may be considered an obstacle. This approach leads to fast implementations, because image columns are analyzed independently. Unfortunately, the slope θ_{1-D} of the 1-D range profile is, in general, smaller than the true slope θ of the visible surface, defined by the angle between the surface normal and the vertical axis. Indeed, it is easy to show that the two are related as

$$\theta_{1-\mathrm{D}} = \arccos \frac{1}{\sqrt{1 + \tan^2 \theta \ \cos^2 \varphi}} \qquad (1)$$

where φ is the angle between the surface normal and the slicing plane Π. This means that, depending on the relative viewing geometry, a sloped surface may not be detected if the surface normal points away from the camera. This characteristic is common to all algorithms that work in a column-by-column fashion using stereo data (Matthies and Grandjean, 1994) or in a scan-by-scan fashion using ladar data (Hong et al., 2000; Batavia and Singh, 2002).

The algorithm described in this paper overcomes this deficiency and computes actual 3-D slopes, while retaining most of the simplicity and computational efficiency of the early approach. We present our detection technique in Section 2.2, after providing in Section 2.1 a simple but rigorous definition of obstacle points that make sense for off-road environments, formalizing and extending the intuitive notion in Matthies et al. (1996). Once the obstacle points have been identified in the image, they can be clustered together (or *segmented*) into isolated obstacles. We show in Section 2.3 that obstacle segmentation (OS) corresponds to finding connected components in a suitable graph built by the OD procedure. Our OS algorithm uses full 3-D information, and thus produces more meaningful results than the simple image-based segmentation procedure of Matthies et al. (1996).

It is instructive to compare our approach with other stereo techniques that detect obstacles by finding surface elements that are above a reference plane (*horopter* (Baten et al., 1998)). Whereas horopter-based obstacle detection is very fast (as it does not require any post-processing of the stereo data), the definition of a "ground plane" is problematic for cross-country environments, as discussed earlier. In addition, the orientation of the horopter with respect to the camera's reference system should be continuously adapted to the current vehicle's attitude. In contrast, our algorithm looks only at local surface slopes, and is more resilient to the vehicle's tilt and roll, as discussed in Section 2.2.1.

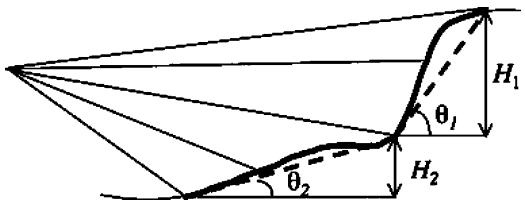

Figure 3. An example of 1-D range profile.

Also note that our algorithm detects obstacles based on a single frame of range data, and therefore does not require integration over multiple frames (with possible additional errors due to image mis-registration and/or inaccurate localization).

2.1. Obstacle Points: Definition

As mentioned in the previous section, a surface ramp is considered part of an obstacle if its slope is larger than a certain value θ and if it spans a vertical interval larger than some threshold H. Unfortunately, surfaces in cross-country environments can seldom be modeled as ramps (i.e., linear patches). This section extends our obstacle definition to arbitrarily shaped surfaces. We assume here that a world reference frame has been defined as in the previous section, and that the range measurement system computes coordinates $(p_{i,x}, p_{i,y}, p_{i,z})$ of visible surface points $\boldsymbol{p}_i$ with respect to this reference frame. We will define obstacles in terms of two distinct points in space as follows:

Definition 1. Two surface points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are called *compatible* with each other if they satisfy the following two conditions:

1. $H_{\min} < |p_{2,y} - p_{1,y}| < H_{\max}$ (i.e., their difference in height is larger than $H_{\min}$ but smaller than $H_{\max}$);
2. $\frac{|p_{2,y}-p_{1,y}|}{\|\boldsymbol{p}_2-\boldsymbol{p}_1\|} > \sin\ \theta_{\max}$ (i.e., the line joining them forms an angle with the horizontal plane larger than $\theta_{\max}$);

where $H_{\min}$, $H_{\max}$ and $\theta_{\max}$ are constant parameters.

Definition 2. Two points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ belong to the same obstacle if:

1. They are compatible with each other, or:
2. There exists a chain of compatible point pairs linking $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$.

Note that these definitions identify obstacles only indirectly: we will say that $\boldsymbol{p}$ is an "obstacle point" if there exists at least one other visible surface point that is compatible with $\boldsymbol{p}$. One potential drawback of this definition is that isolated points protruding from the ground may not be classified as obstacles (if they admit no compatible points). However, such isolated points are usually measurement errors, which would probably be discarded anyway. The condition involving $H_{\max}$ enforces separation of two obstacles in those cases where pairs of points exist, one for each obstacle, satisfying the slope condition but located far apart (see Sections 2.2.1 and 2.3). The compatibility relationship can also be expressed as follows (see Fig. 4):

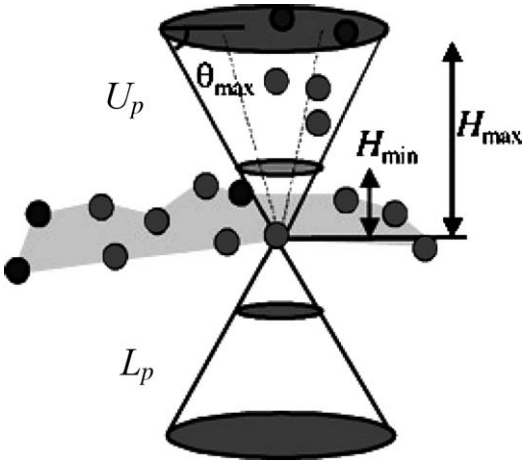

Figure 4. The two truncated cones U_p and L_p for the determination of compatible points.

Equivalent Definition 1. The points compatible with a surface point $\boldsymbol{p}$ are those belonging to the two truncated cones U_p and L_p with vertex in $\boldsymbol{p}$, axis oriented vertically, and limited by the two planes of equation $y = H_{\min}$ and $y = H_{\max}$ respectively.

2.2. Obstacle Detection (OD) Algorithms

To find all obstacle points in a range image, we need to look for pairs of compatible points. A naïve strategy would examine all point pairs, resulting in $N^2 - N$ tests, where N is the number of points in the image for which range was computed. In fact, it is not necessary to test all point pairs, but only a much smaller subset of them. For if $\bar{\boldsymbol{p}}$ is the projection of a 3-D point $\boldsymbol{p}$ onto the image plane, then the points compatible with $\boldsymbol{p}$ must project[2] onto two truncated triangles ($U_{\bar{\boldsymbol{p}}}$ and $L_{\bar{\boldsymbol{p}}}$) with vertex in $\bar{\boldsymbol{p}}$ (assuming zero tilt and roll; for non-zero tilt/roll the triangle projections change, as discussed in Section 2.2.1). These are approximately isosceles triangles (see Fig. 5), with vertical symmetry axis, height equal to $H_{\max} f / p_z$ (where f is the camera's focal length), and base approximately equal to $\frac{2H_{\max} f}{\tan\theta_{\max} p_z \cos\upsilon}$ where $\upsilon = \arctan p_x/p_z$ is the

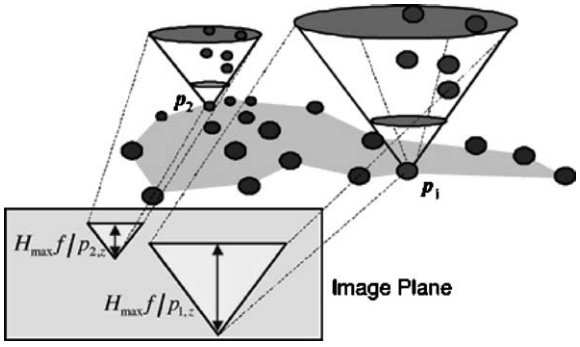

Figure 5. The projection of the truncated cones onto the image plane (only upper cone shown)

azimuth angle of $\boldsymbol{p}$ measured on the x-z plane from the z axis (which can be derived directly from $\bar{\boldsymbol{p}}$ and the intrinsic camera parameters). This observation suggests the following algorithm, with complexity linear in N:

OD Algorithm 1.

- Initialization: Classify all pixels as non-obstacle.
- For each pixel $\bar{\boldsymbol{p}}_i$:
  - Scan the points in the two truncated triangles $U_{\bar{p}_i}$ and $L_{\bar{p}_i}$ until a pixel $\bar{\boldsymbol{p}}_j$ compatible with $\bar{\boldsymbol{p}}_i$ is found, in which case classify $\boldsymbol{p}_i$ as an obstacle point and move on to the next pixel.
  - If no pixel in $U_{\bar{p}_i} \cup L_{\bar{p}_i}$ compatible with $\bar{\boldsymbol{p}}_i$ is found, $\boldsymbol{p}_i$ is not an obstacle point.

One drawback of this algorithm is that it may test the same point pair $(\boldsymbol{p}_i, \boldsymbol{p}_j)$ twice. The overall number of tests can be estimated as follows. Let α be the expected proportion of obstacle points in the image, and let K be the average number of points in each projected truncated triangle on the image plane. The average number of tests for each non-obstacle point is thus $2K$. If one assumes that there is a proportion $\alpha = L/K$ of obstacle points within each double truncated triangle, then the probability of finding the first obstacle point after exactly n tests is

$$P_n = P(\text{obstacle point found at } n \text{ - th test}) = \begin{cases} (1-\alpha)^{n-1}\dfrac{2L}{2K-n-1}, & 1 \le n \le 2(K-L)+1 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

No simple expression can be found for the expectation of n in (2). Instead, we may approximate it with a geometrically distributed variable:[3] $\bar{P}_n = (1-\alpha)^{n-1}\alpha$, which has expectation equal to $1/\alpha$.

Thus, the expected overall number of tests is approximately $N(1 + 2K(1-\alpha))$.

Let us now introduce a second strategy, which does not require duplicate tests:

OD Algorithm 2.

- Initialization: Classify all pixels as non-obstacle.
- Scan the pixels in the image from bottom to top and from left to right. For each pixel $\bar{\boldsymbol{p}}_i$:
- Examine *all* points in $U\bar{\boldsymbol{p}}_i$, and determine the set S_i of points compatible with $\bar{\boldsymbol{p}}_i$.
- If S_i is not empty, classify all points of S_i as well as $\boldsymbol{p}_i$ as obstacle points.

It is easy to see that each pixel $\bar{\boldsymbol{p}}_i$is tested only once against all other pixels in $U_{\bar{p}_i} \cup L_{\bar{p}_i}$. With reference to the quantities introduced earlier, now NK tests must be performed over the image. Thus, if the expected ratio of obstacle points αgn the image is less than $0.5(1 + 1/K)$, the second algorithm results in higher computational efficiency.

2.2.1. Parameter Selection. Our OD algorithms rely on a set of parameters (the slant angle threshold θ and the minimum/maximum height thresholds $H_{\min}$ and $H_{\max}$). This section provides some general guidelines for selecting suitable parameter values.

The slant threshold should correspond to the maximum slant that the vehicle can safely negotiate. All the experiments of this paper use $\theta = 40^\circ$. The minimum height threshold $H_{\min}$ should be less than or equal to the vertical clearance under the vehicle ($H_{\min} = 0.2$ m in our experiments). Note that, whereas $H_{\min}$ determines the minimum size of a detected obstacle, two points in the same obstacle may be separated by a vertical distance smaller than $H_{\min}$, as long as they are linked by a chain of compatible point pairs.

Differently from $H_{\min}$, the maximum height threshold $H_{\max}$ is determined by the quality (density) of the stereo range estimates. The choice of $H_{\max}$ should be based on the following two observations. A small value for $H_{\max}$ increases the risk of over-segmentation (a single obstacle gets identified as two distinct obstacles vertically stacked) if there is a "hole" in the measured point cloud. On good quality stereo data the point cloud will

be dense and therefore in such cases $H_{\max}$ can be selected to be equal to $H_{\min}$. This will signficantly speed up the run-time of the obstacle detection algorithm. On data sets with medium-poor quality stereo, on the other hand, a large value for $H_{\max}$ is necessary to handle missing segments in the point cloud. However, this increases the computational load of the algorithm, since $H_{\max}$ determines the size of the search window. Furthermore, according to Definition 1, $H_{\max}$ determines the maximum horizontal distance $D_{\max}$ for two points to be compatible (it is easy to see that $D_{\max} = 2H_{\max} \cot\ \theta_{\max}$). By limiting $D_{\max}$ to a value smaller than or equal to the vehicle's width, one ensures that the algorithm does not mistakenly merge two obstacles when there is enough horizontal clearance to drive between them. In our tests, we used $H_{\max} = 1$ m.

It should also be noted that our "canonical" hypothesis that the y axis of the chosen reference frame always points vertically is not very realistic. Normally, the camera reference frame is attached to the vehicle, hence when the vehicle itself is tilted our original hypothesis fails. If the actual vertical direction can be recovered (based on IMU information), it is in principle possible to adaptively re-adjust the camera reference frame to account for the new attitude. This simply amounts to pre-multiplying the values of coordinates of the measured 3-D points by a suitable rotation matrix. Note that the search windows of OD Algorithms 1 and 2 (the truncated projected triangles discussed earlier) will also change when the vehicle tilts. If the vehicle tilts by angle θ, the height of each projected triangle will change by a factor of approximately $\frac{1}{\cos\gamma \pm \frac{H_{\max}}{p_z}\sin\gamma}$, while the base of the triangle projection will be approximately unchanged. For a pure rotation ϕ, the projected triangles will rotate by the same amount ϕ on the image plane. For combined rotation and tilt, the projected triangles will rotate and their lengths and heights will change. We compensate for small vehicle tilt and roll by using a larger projected triangle search window than the theoretically computed parameters. This obviates the need to estimate vehicle state information from IMU or visual odometry.

2.3. *Obstacle Segmentation*

Besides detecting obstacle points, it is useful to identify all distinct obstacles in the image. This facilitates further reasoning about obstacle size and type, and inhibits undesired "gap filling" if the measured 3-D points are interpolated.

In our earlier work (Matthies et al., 1996; Bellutta et al., 2000), obstacle segmentation corresponded to finding the connected components of the obstacle pixel map on the image plane. This approach is computationally simple, but may incorrectly link two distinct obstacles when their projections on the image plane overlap, even though they are well separated in space (under-segmentation). Additionally, this prior solution may assign more than one label to a single obstacle when it has gaps or missing range points (over-segmentation). This can be avoided by using the equivalence relationship stated in Definitions 1 and 2, which makes full use of 3D spatial information.

It is useful to associate each measured point with one node of an undirected graph (*point graph*), where two nodes are linked by an edge if the corresponding points are compatible. Thus, according to Definition 2, two points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ belong to the same obstacle if and only if there exists a path in the graph from $\boldsymbol{p}_1$ to $\boldsymbol{p}_2$. We can extend this notion to define a single obstacle as a maximal connected subgraph (i.e., a connected component) of the point graph. The connected components of the point graph may be computed online at the same time of graph creation (i.e., within OD Algorithms 1 or 2) using the Union—Find algorithm (Mehlhorn, 1984).

Note that our obstacle segmentation algorithm does not require pixels colored with the same label to be spatially connected (in 2D image space) to each other. Indeed, the algorithm is able to correctly assign only one label to a single obstacle even when there are gaps in the range measurement. This is an important advantage in scenes with sparse stereo range data.

2.3.1. Shape-Based Validation. Our obstacle detection algorithm identifies a number of distinct obstacles as distinct sets of image pixels. It is often the case that, due to measurement noise, a number of such clusters need to be rejected as spurious. Previous work (Bellutta et al., 2000) used 2-D area information (such as region areas) of pixel clusters in the image plane as a criterion for rejection, the rationale being that very small connected components are likely to be "outliers" and therefore should be discarded. However, it should be clear that, due to perspective, 2-D region area is not a very significant measure. Given that the spatial coordinates of the obstacle points are known, it makes more sense to define meaningful 3-D attributes of shape and volume. Our current approach builds simple attributes from the raw point-cloud data, enabling real-time analysis. Some of the features we have experimented with

are the volume of the 3-D bounding box around an obstacle, the average and maximum slope, as well as the obstacle height. A simple rule-based system (assigning thresholds to all such attributes) rejects obstacles with small bounding volume, small average/maximum slopes, or small average/maximum relative height. For example, all detected obstacles with an average slope lesser than 2.5, or maximum slope lesser than 5.0 are considered false positive.

2.3.2. Experimental Results. Our obstacle detection algorithm was implemented and prototyped in C/C++. Initial implementation of the code ran at 1.5 Hz to 0.5 Hz on 320×240 images, on a 900 Mhz Pentium III PC. The variable speed of the algorithm resulted from differences in density (quality) of stereo-range data in the image, the window size used in the algorithm for nearby and distant pixels, and camera orientation and placement on the vehicle (which determines the number of visible ground points).

Some detection and segmentation results for typical cross-country situations are shown in Figs. 6 and 7 (the images shown in the figures correspond to the left camera of the stereo pair). Stereo can only compute disparities where the left and right images in the pair overlap; hence, the leftmost columns of the image contain no range measurements (see Fig. 6(b) and (c)). Note that our technique does not compute the ground plane, and thus can be used successfully with cases such as the last image of Fig. 7, where no ground plane can be defined.

The images in Figs. 6 and 7(a) and (c) contain slanted surfaces pointing away from the camera, which represents a challenge for the columnwise algorithm of Matthies et al. (1996) and Bellutta et al. (2000) but not for our 3-D based algorithm. The image in Fig. 7(b) contains a "negative obstacle" (a ditch across the road). A method to detect negative obstacles based on range information is discussed in Bellutta et al. (2000). Our algorithm correctly detects the visible face of the ditch as an obstacle even though it is below the ground level.

3. Terrain Classification

As mentioned in the Introduction, terrain classification is a required capability for efficient navigation in

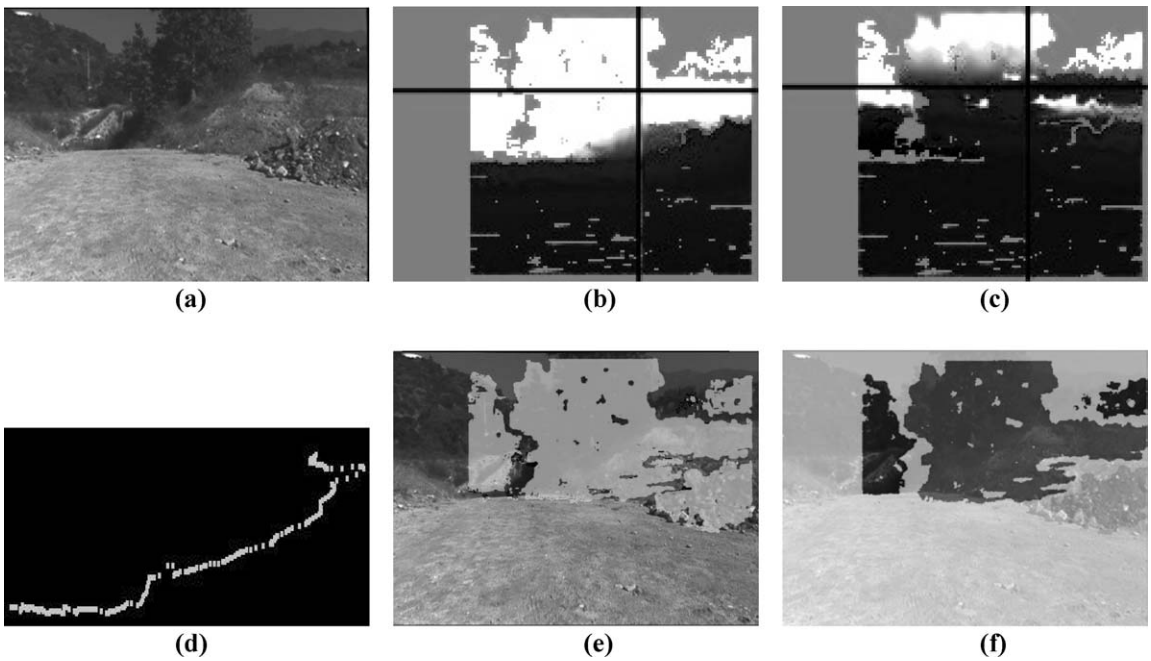

Figure 6. The measured range map (b) and elevation map p_y (c) for the image in (a). The elevation profile for the image column highlighted in (b) and (c) is shown in (d). More precisely, (d) shows the elevation as a function of the distance from the camera for one slicing plane (see Fig. 2). The computed obstacle points and ob-stacle segmentation are shown in (e) and (f) respectively.

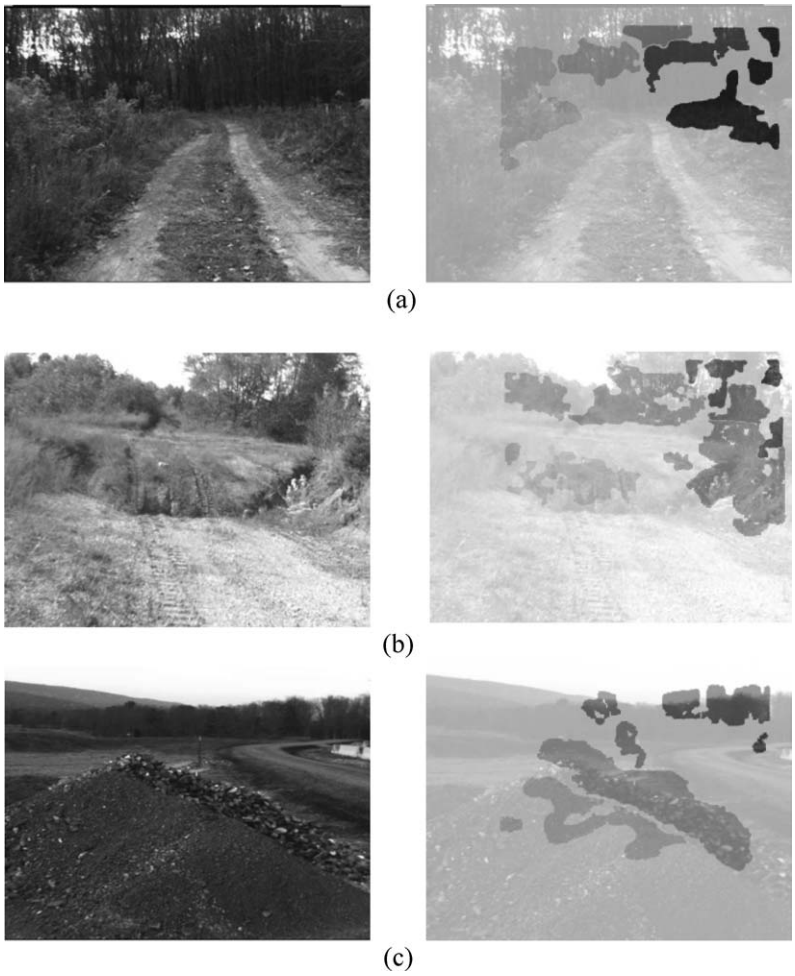

Figure 7. Examples of obstacle detection and segmentation. Different obstacles are shown with different gray values in the right column.

vegetated environments (Bellutta et al., 2000; Hebert et al., 2002). Information about the terrain composition, in conjunction with the geometric description of visible "obstacles", can be used for selecting the optimal path as well as the optimal velocity for traversal. More precisely, given the elevation profile along a candidate path, and some information about the compressibility of terrain cover, one may predict the dynamic behavior of the vehicle as it drives through that path at a given speed. Vegetation such as grass or bushes is compressible, therefore the visible surface does not correspond to the "load bearing" surface (where the vehicle's tires actually touch the ground). Simple models for the vehicle's dynamics in vegetated terrain of known compressibility characteristics were presented in Talukder et al. (2002), based on the earlier model for bare (uncompressible) soil of Rankin et al. (1998). In this paper we concentrate on the perception problem, using sensors normally found on autonomous robots: color cameras and laser rangefinders (ladars). Both sensors have advantages and disadvantages. Color cameras are passive, which has tactical value. They are cheap, small and light, produce full frame data at an acceptable rate, and can be used to compute range by stereopsis. Unfortunately, regular cameras cannot be used at night, and the quality of stereo data is not always satisfactory, especially in the case of grass patches and dense vegetation. Ladars, on the other hand, can be used by day and night, and usually produce very accurate and dense range data. The interpretation of range profiles for terrain typing is perhaps less intuitive than color analysis, but is still feasible, as we show in Section 3.2 (see

also Macedo et al., 2000; Hong et al., 2000). However, ladars (especially imaging ladars) are bulkier, heavier and more expensive than cameras.

We discuss techniques for terrain classification using color images and ladar data in the next two sections. Note that the chosen class taxonomy depends mainly on two factors: which classes are useful for autonomous navigation, and which classes can actually be detected with the available sensor information. The right balance must be struck between the desire for high descriptiveness (which would lead to dense taxonomies) and the need for robust classification (which favors fewer classes). In the case of color-based classification (Section 3.1), we normally consider the following set of classes: soil/rock, green (photosynthetic) vegetation, dry (non-photosynthetic) vegetation (which includes tree bark), as well as a "none of the above" class. For the case of range analysis from a ladar (Section 3.2), we consider discrimination of "obstacles" characterized by relatively smooth surfaces (such as rocks or tree trunks) from grass or foliage.

3.1. Color-Based Classification

The apparent color of a surface is a function of the irradiance spectrum on the camera, which, in clear atmospheric conditions, is a combination of the radiance spectrum of the illuminant(s), the reflectance spectrum of the surface, and the relative geometry of the illuminant and of the observer with respect to the surface (Horn, 1986; Elachi, 1987). Of all these components, it is the surface reflectance that carries information about the material type. Examples of reflectivity spectra within the visible and Near Infrared (NIR) band for three materials of interest (green or photosynthetic vegetation, dry or non-photosynthetic vegetation, and soil) are shown[4] in Fig. 8. Visible wavelengths cover the interval from approximately 400 nm to 700 nm. NIR wavelengths up to about 1 μm are not visible to human eye but are within the range of CCD sensors (although almost all cameras have an optical filter to block such wavelengths out). It can be noted from Fig. 8 that green vegetation is easily separable from the other two classes, both in the visible and in the NIR band. In particular, its reflectance has a characteristic feature at approximately 750 nm (beyond the visible spectrum), sometimes called the "red edge" (Elachi, 1987). Soil reflectance curves generally exhibit a gentle increase with increasing wavelength. In particular, soil reflectance depends upon the chemical and physical properties of the components, moisture content (wetter soils appear darker) and texture, and surface roughness. From Fig. 8, one also notes that the spectrum of dry vegetation appears very similar to that of soil. This makes color-based discrimination between the two classes rather challenging in some cases, and represents a limitation of this method (Roberts et al., 1993).

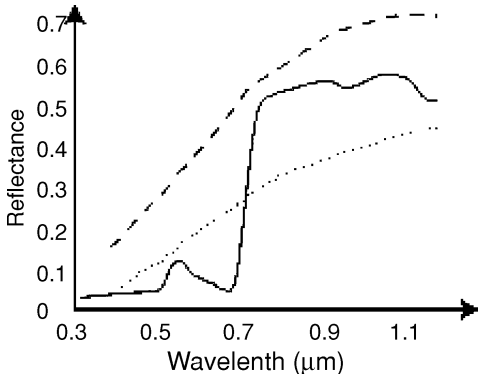

Figure 8. Typical reflectance curves for photosynthetic vegetation (solid line), soil (dotted line), and non-photosynthetic vegetation (dashed line) (from [17]).

3.1.1. Normalization and Calibration. Ideally, one would like to derive from the measured $[r, g, b]$ color, the "normalized" surface color (that is, the color of the surface under canonical illumination and observation conditions, which is a function only of the surface reflectance). Brightness normalization (Healey, 1992), which transforms the vector $[r, g, b]$ into $[r, g]/(r + g + b)$, is a simple and popular approach. The new vector is invariant to scalar transformations of the type $\alpha[r, g, b]$, which can model the effect of a number of variables: the intensity of the illuminant, the angle between the surface normal and the illuminant direction (under the Lambertian hypothesis (Horn, 1986), and the iris aperture of the camera. However, the brightness component conveys information about the surface reflectance that is useful for discrimination, and in practical cases this procedure does more harm than good, at least when enough data is available to train the system. A more useful type of normalization is white point calibration, a standard procedure (usually already implemented in the camera hardware) that yields a diagonal transformation of the color vectors (i.e. of the type $[\alpha r, \beta g, \gamma r]$, where α, β, γ are positive coefficients). White point calibration attempts to normalize the perceived colors with respect to the

spectral composition of the illuminant. The coefficients are computed by imaging a white surface, and forcing the transformed color vector into a canonical triplet. The procedure can be repeated periodically, or every time the illuminant changes. Apart from the burden of presenting a white surface patch to the camera, white point calibration has two main drawbacks. First, the diagonal transformation is only an approximation of the actual change of perceived color of surfaces under a changing illuminant (Maloney and Wandell, 1986). Second, white point calibration assumes that only one illuminant type is present in the scene. This is not true in normal outdoor environments, which usually contain at least two illuminants (direct sunlight and diffuse light or shade). Despite these shortcomings, white point calibration is a viable pre-processing stage that helps improve classification.

Color constancy techniques that use more complex models of color formation for illuminant compensation have been proposed (for a review, see Buluswar and Draper, 2002). However, these methods often rely on unrealistic assumptions about the scene or require laborious procedures to estimate the reflectance and illumination parameters. In contrast to model-based techniques, we adopt an exemplar-based approach, and learn color distributions of terrain classes by training over a large number of images taken under widely different illumination conditions. In typical field experiments, we collected training data containing surfaces of interest under direct sunlight and in the shade, and at different times of the day. Extensive testing has shown that this method works very well for outdoor images, provided that (1) a representative training data set can be collected, covering all expected environmental conditions, and (2) the classifier is able to adequately represent the variability of perceived color within each single class. In the next section, we discuss a simple Maximum Likelihood classifier that has given good results for this task, based on Mixture of Gaussians modeling.

3.1.2. Classification Using Mixtures of Gaussians. We use a maximum likelihood (ML) strategy for pixel-wise classification. For a given class k, the perceived color c is modeled as a random vector with density (class-conditional likelihood) $f(c \mid k)$. Each pixel is thus assigned to the class that maximizes $f(c \mid k)$. Note that if prior probabilities $P(k)$for the classes are set, then ML classification is easily converted into MAP classification, by choosing the class k that maximizes $P(k) f(c \mid k)$. Our classifier models the conditional class likelihood for a given class k with a Mixture of Gaussians:

$$f(c \mid k) = \sum_{j=1}^{M(k)} \alpha_{k,j} G(c; \mu_{k,j}, \Sigma_{k,j}) \tag{3}$$

where $G(c; \mu_{k,j}, \Sigma_{k,j})$ is a Gaussian density with mean $\mu_{k,j}$ and covariance $\Sigma_{k,j}$, and $\alpha_{k,j}$ are the mixing coefficients (with $\alpha_{k,j} \geq 0$ and $\sum_j \alpha_{k,j} = 1$) (Bishop, 1995). The quantity $M(k)$ represents the number of Gaussian clusters (or modes) within the given class k. The model parameters $\{\alpha_{k,j}, \mu_{k,j}, \Sigma_{k,j}\}$ can be learned from labeled training images[5] using the Expectation Maximization (EM) algorithm (Bishop, 1995). The starting point for the EM algorithm is obtained by an initial k-means clustering (Bishop, 1995), itself initialized by random seeds. The ability of the model to represent multi-modal densities is indeed very important for outdoor images. For example, for the same terrain type, the color distributions under direct light and under diffuse light (shade) usually correspond to different Gaussian modes. There exist a number of standard techniques in the literature to select the "correct" number of modes $M(k)$ for each class (McLachlan and Peel, 2000, chapter 6), none of which, in the authors' experience, is really satisfactory. Choosing too few modes leads to inaccurate representation, while too many modes may overfit the data. In practice, a number of modes between 3 and 5 proved a good compromise in most cases.

In addition to the chosen set of terrain classes, it is important to also provide a "none of the above" (or "outlier") class, to which all of the pixels that do not correspond to any of the selected classes should be assigned. Intuitively, these are colors that are not well explained by the chosen model. As explained in Ripley (1996), we select outliers by analyzing the total likelihood of the color:

$$f(c) = \sum_{k=1}^{K} P(k) f(c \mid k) \tag{4}$$

where $K$is the number of classes, and $P(k)$ is the prior probability of class k (for example, in the case of uniform priors, $P(k) = 1/K$). Thus, a color c is assigned to the outlier class if $f(c) < f_0$, where f_0 is a threshold constant. The choice of f_0 may be problematic: the total likelihood is a probability density function, and there is no intuitive criterion to guide the threshold selection. Following (Ripley, 1996), we link the threshold f_0 to the probability P_0 that a color pixel will be assigned to

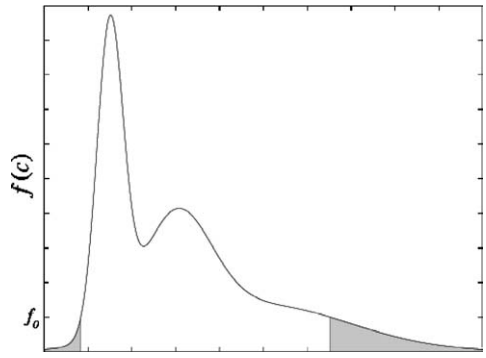

Figure 9. The pdf of a mixture of 3 Gaussians, and the outlier threshold f_0. The area of the shaded region is equal to P_0 in (5).

the outlier class based on our color model (see Fig. 9):

$$P_0 = \int_{f(c)<f_0} f(c)\,dc \tag{5}$$

P_0 (a monotone function of c) is a more intuitive parameter to use than f_0, since it represents an actual probability value. Given P_0, f_0 can be found by solving (5). For any candidate value of f_0, the integral on the right hand side of (5) is computed by standard Monte Carlo sampling. Then, standard nonlinear 1-D root finding techniques can be used to solve the integral Eq. (5).

Figure 10 shows the classification result on an image containing objects (the red poles) that never appeared in the training data.[6] The classifier correctly assigns those surfaces to the outlier class (color-coded in black in the classification map). The threshold f_0 was determined by solving Eq. (5) for $P_0 = 0.99$. Note that also the pixels corresponding to the sky (which was not used for training) are classified as outliers.

Figure 11 shows two examples of color-based classification in outdoor scenes. Only those pixels for which the range (as measured by stereo) was less than 50 meters were classified. It is not particularly useful to try to analyze terrain at longer distances, and chromatic shift may occur due to atmospheric effects (Nayar and Narasimhan, 2000). These images did not belong to the training data set. It should be noted that in the first image, the system correctly classified the scene under direct sunlight as well as in the shade. The second image in Fig. 11 highlights the inherent color ambiguity between the "soil" and the "dry grass" classes, discussed in Section 3.1. Indeed, most of the surface of the mound of ground to the right of the image was incorrectly classified as dry grass. Unfortunately, this ambiguity cannot be removed based on color alone (Roberts et al., 1993). Some other features (such as visual texture (Castano et al., 2001; Shi and Manduchi, 2003) or thermal multispectral analysis (Abedin et al., 2003)) could be used jointly with color to enhance the classifier performance in these cases.

3.1.3. Open Problems. In our experience, color is a very effective feature for terrain typing. Still, there are a number of open problems related to color that require further investigation. Two such problems, which are the object of current research in our laboratories, are briefly summarized in the following:

1. *Illuminant compensation.* As mentioned earlier, traditional color constancy algorithms are not very effective for automatic illuminant compensation in outdoor scenes. Our exemplar-based approach trains the system with a plethora of images taken of the scene under as many illumination conditions

Figure 10. Color-based classification. The color code for the classification map in the right image is the following: Light gray: soil/rock; medium gray: dry vegetation; dark gray: green vegetation; black: outlier.

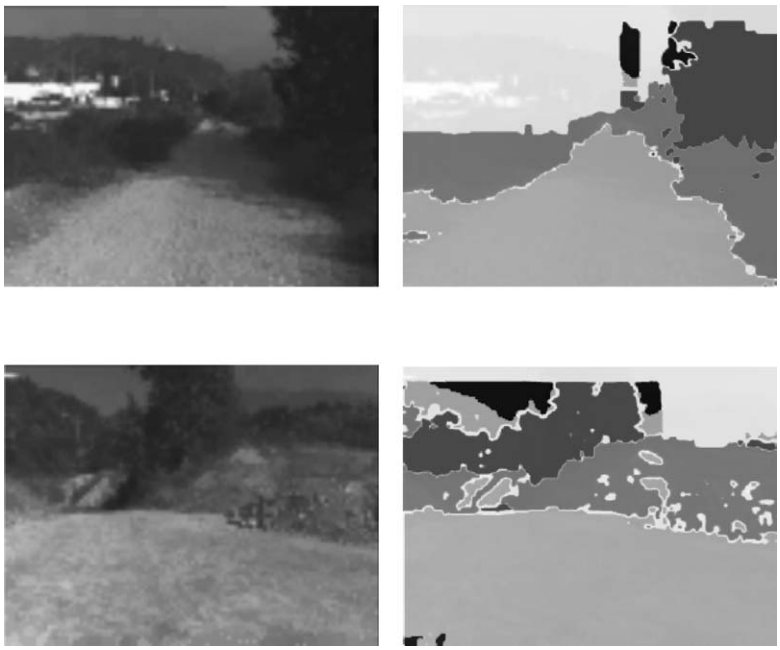

Figure 11. Color classification examples. The classification maps in the right image use the same grayscale code as in Fig. 10. Only points for which the range was less than 50 meters were classified.

as possible. While this approach has proven very reliable in our experiments, it requires a substantial training data collection and labeling effort that can make it unpractical. In Manduchi (2004) we present preliminary results using a technique that allows one to train the system with only one (or a few) images containing the different types of terrain of interest under just one "canonical" illuminant. The knowledge of the color statistics of each class under the canonical illuminant enables the system to estimate the illuminant (or illuminants, such as direct sunlight and shade) contained in a new scene at the same time as classification. A Maximum Likelihood strategy is used to assign illuminant labels to points in the scene, as well as to determine the illuminants' parameters. Prior knowledge of the illuminant parameters' distribution can be also used in a Bayesian framework, in order to improve the robustness of the system. The interested reader is encouraged to consult (Manduchi, 2004) for more details.

2. *Color shift in atmospheric conditions*. It is well known that, under some specific adverse atmospheric conditions (such as fog or haze), the perceived color of a surface changes according to the distance of the surface to the observer. The physics of this phenomenon is well understood (McCartney, 1976; Nayar and Narasimhan, 2000). Indeed, color shift can be a significant nuisance for outdoor color classification; our current research is investigating strategies for recovering the original surface color based on the estimated atmospheric parameters and the distance to the visible surfaces (which can be computed, for example, by stereopsis).

3.2. *Ladar-Based Classification*

The accurate range profile produced by a ladar can be used to discriminate between certain classes of terrain. In this section we use ladar data to discriminate between grass and other types of surfaces that are likely to belong to obstacles. Although OD is usually studied as a geometric identification problem (as in Section 2), we show here that in environments characterized by tall and dense vegetation OD also requires terrain cover classification. Indeed, geometry-based obstacle detection cannot be used directly in these environments, because grass tufts would be perceived as obstacles. Our goal is thus to discriminate grass patches from non-traversable obstacles such as rocks and tree trunks.

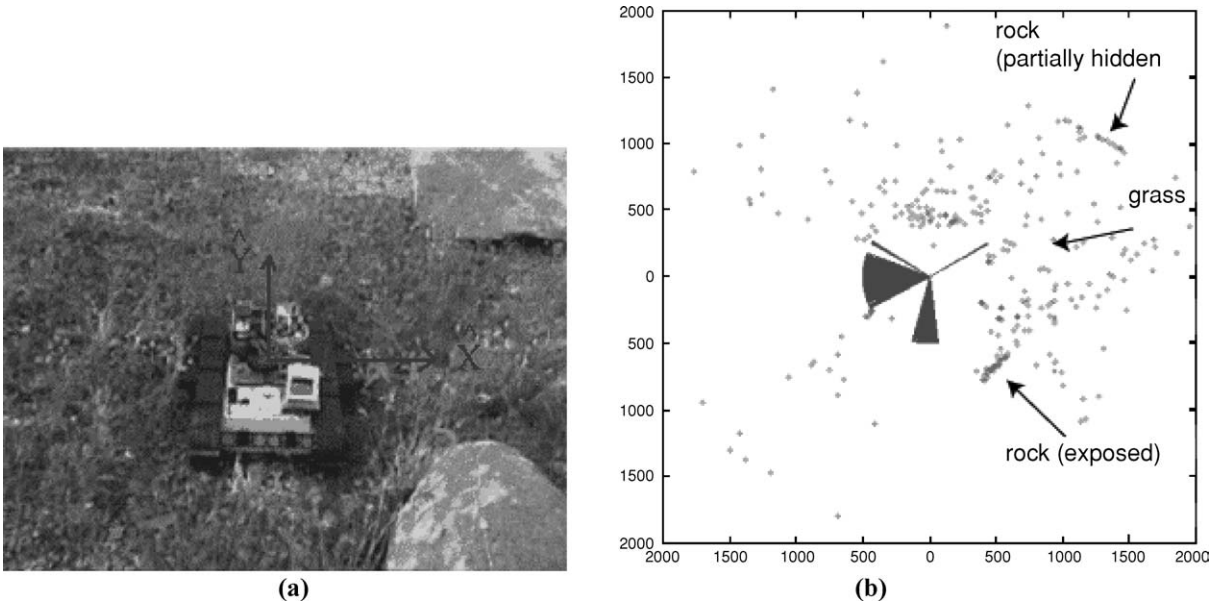

Figure 12. (a) The "Urbie" robot facing two rocks at 45° and 315°. (b) The measured range points (the axis units are in millimeters). The fan-shaped clusters of lines correspond to directions with no range estimation due to self-occluding structures on the robot.

Range data is well-suited to this type of recognition. Intuitively, one would expect that range data should be spatially scattered in areas that correspond to grass and bushes, and should exhibit a more regular and smooth pattern on bare soil or rock (Macedo et al., 2000). Thus, local analysis of range profiles should provide enough information to discriminate between these two cases. An example of a range profile taken by a ladar system rotating around the vertical axis is shown in Fig. 12(b). Some ladars are capable of returning both range estimates and the intensity of the returns associated with each range estimate; we do not use intensity information as our interest is to develop a range-only approach that could be extended to the analysis of cloud of points not necessarily obtained via ladar. Our ladar, described in detail in the next section, was mounted on the Urbie robot, a small Packbot designed for urban environments, shown in Fig. 12(a). The dots shown in Fig. 12(b) correspond to the laser hits for the scene shown in (a). The fan-shaped clusters of lines indicate directions that were not analyzed in the scan because there was a permanent occlusion in the robot that prevented the laser from being fired in that direction (such as a camera mounted to the left of the laser occluding the ladar's view to the left of the robot).

In the following sections, we present an algorithm that can reliably discriminate between grass and obstacle surfaces based in the analysis of the range from a single-axis ladar (Castano and Matthies, 2003). An attractive feature of this algorithm is that it is able to detect surfaces when they are partially hidden by the vegetation, such as in Fig. 12. We describe in Section 3.2.1 the characteristics of the ladar used in our system. Statistical features of range data in grass are discussed in Section 3.2.2. The properties derived in Section 3.2.2 are instrumental in the analysis of our obstacle detection algorithm, described in detail in Section 3.2.3.

3.2.1. The Ladar System. We consider here a ladar collecting range measurements as it rotates around a vertical axis. In particular, the examples in this paper are based on data from a customized Acuity AccuRange 4000 (Bergh et al., 2000; Matthies et al., 2000). This ladar operates at a wavelength of 780 nm, in the NIR domain, rotates at a rate of 5 Hz and can estimate range up to distances of 15 m (50 ft). The ladar is capable of acquiring up to 1024 samples per revolution but we set the sampling interval to 10 mradians (i.e., $\Delta\alpha = 0.57°$) to relax a constraint on the minimum size of the grass blades, as explained later.

The accuracy of each range estimate is affected by parameters such as distance, environmental temperature and color and pose of the target (Adams, 2000). The standard deviation σ_d of the estimation error is a

function of the measured range d, as well as of the angle of incidence. For our experiments, we set the operative range to [0.3 m, 2.0 m].

3.2.2. Statistics of Range in Grass. To better justify our detection algorithm, we introduce some statistical models for the measured range in situations of interest. For simplicity's sake, our analysis assumes that the measurements are noiseless.

We begin with the simple case of a linear surface patch, corresponding to an "obstacle" (Note that in context of this section, "obstacle" is any object that is not grass.) Consider a small angular span $\alpha \in [0, A]$, and assume that the surface is at angle γ with the laser beam at $\alpha = 0$ (see Fig. 13), where the measured range is $d(0) = D$. The measured range at angle α is thus

$$d(\alpha) = D \sin(\gamma)/\sin(\alpha + \gamma) \tag{6}$$

For small angular spans, we can linearize Eq. (6) and write

$$d(\alpha) = d(0) + \alpha \left.\frac{\mathrm{d}d}{\mathrm{d}\alpha}\right|_{\alpha=0} = D(1 - \alpha K) \tag{7}$$

where $K = \cot\gamma$. If the angle α is a random variable uniformly distributed in [0, A], one easily sees that the measured distance, being a linear function of a uniform random variable, is itself uniformly distributed within $[D, D(1 - AK)]$.

Our goal here is to determine the distribution of laser returns in a situation with uniform grass. To simplify our treatment, we will make the following assumptions, which don't significantly affect the validity of our model:

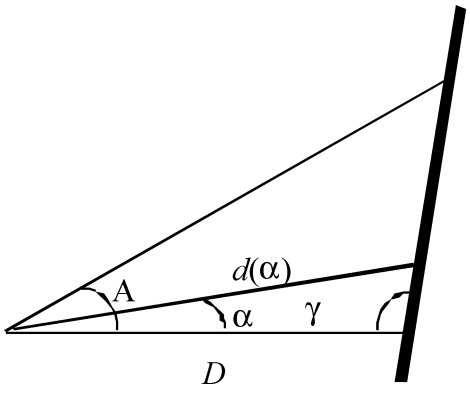

Figure 13. Overhead view of a slanted linear patch, corresponding to an obstacle.

1. The laser beam is infinitesimal, with null divergence;
2. No two laser beams hit the same grass blade;
3. The grass blades in a homogeneous patch have identical circular sections of diameter h, and their centers are distributed as a uniform Poisson field with intensity $\bar{\lambda}$.

Let us briefly comment on such assumptions. Assumption 1 is not consistent with the fact that a laser beam has non-null divergence, and that it has non-null initial diameter. However, finite divergence has a small effect on our analysis, as discussed in Macedo et al. (2000), therefore we can safely neglect it. Assumption 2 is acceptable if the grass blades are thin enough, i.e., if the blade diameter h is smaller than $d \cdot \tan(\Delta\alpha)$ for all distances d within the operative range. In our case, for an operative range of [0.3 m, 2.0 m] and a sampling interval of $\Delta\alpha = 10$ mradians, the constraint is satisfied as long as the blades' diameter h is less than 3 mm in the near field and less than 2 cm in the far field. Assumption 3 is a working hypothesis that we found empirically to be reasonably accurate in typical outdoor situations. A similar hypothesis was used in Leung and Malik (1997), Huang et al. (2000), albeit in a different context.

Based on these assumptions, it can be shown (Leung and Malik, 1997; Huang et al., 2000; Macedo et al., 2000) that the measured noiseless range for the grass case is distributed as an exponential random variable with parameter $\lambda = \bar{\lambda}h$ (as shown in Fig. 14(a), solid line). Figure 15(b) shows the histogram of actual range measurements over a number of revolutions of the ladar in the case of tall homogeneous grass. The grass patch was at a distance of approximately 1.5 meters from the robot, as shown in Fig. 15(a). The best-fitting exponential (superimposed on the histogram in Fig. 15(b)) has parameter $\lambda = 3.7$/m. From visual inspection, it can be noted that the exponential curve approximates the histogram tail rather well.

Suppose now a linear surface patch, such as the one depicted in Fig. 13, is partially hidden by grass. In this case, the maximum possible range at a particular angle α is equal to the distance to the surface $d(\alpha)$. However, the laser beam may be intercepted by a blade of grass before reaching the surface. The analytical expression of the pdf of range within an angular sector A, assuming that the surface is linear and partially occluded by uniform grass, is derived in the Appendix. The shape of this density is shown in Fig. 14(a) (dashed line). It is

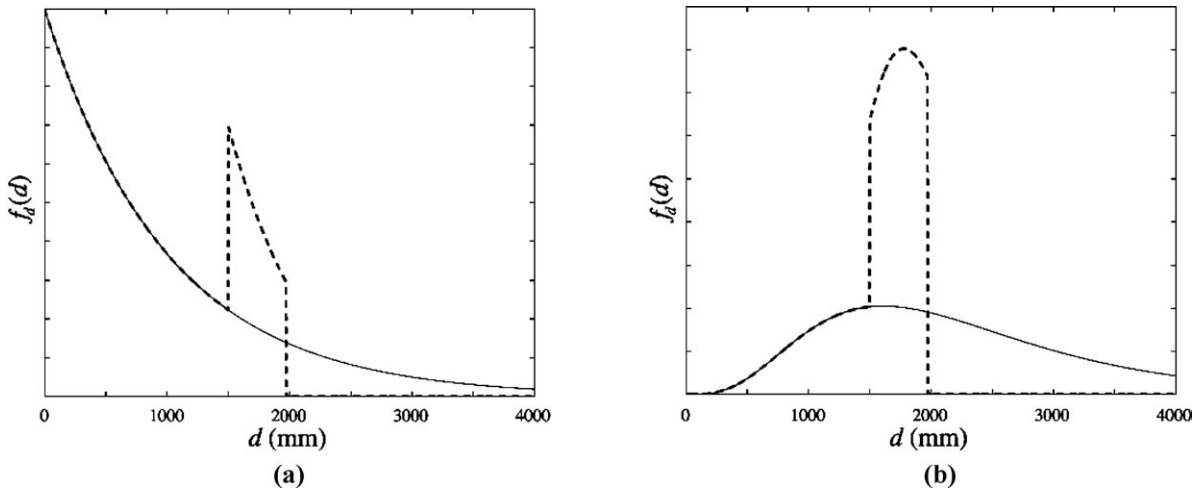

Figure 14. (a) The pdf of the range in the case of grass (solid line) and of a partially hidden obstacle (dashed line). (b) The pdf of the same variables after maximum-value filtering ($n = 5$).

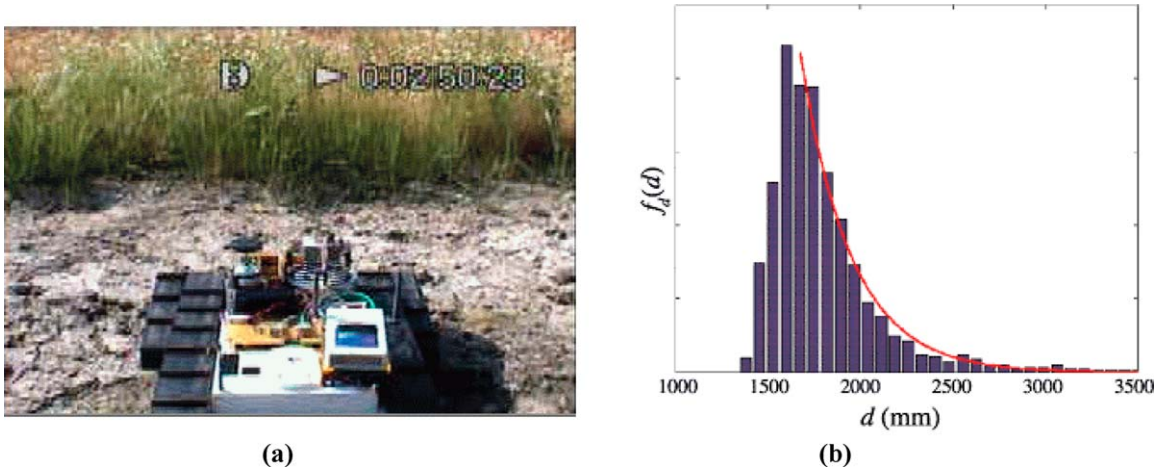

Figure 15. Normalized histogram of range measured over 30 revolutions (b). The curve represents the best-fitting exponential density. The robot was placed about 1.5 m from a patch of tall grass (a).

seen that for values smaller than the minimum distance D to the surface, the range behaves as an exponential variable (like in the uniform grass case); as expected, its pdf vanishes for values beyond the maximum distance $D_{\max}$ to the surface within the sector A (in Fig. 13, $D_{\max} = d(\mathrm{A})$).

3.2.3. The Detection Algorithm. In the previous section it was shown that the presence of a surface, albeit partially hidden, may change the distribution of the measured range in grass. Based on this observation, an earlier detection algorithm (Macedo et al., 2000) used higher order statistics of range, computed over small angular sectors, as discriminative features. However, this algorithm did not exploit the "spatial coherence" property of typical smooth surfaces. In other words, it did not use the observation that if a measurement $d(\alpha)$ comes from an object surface, the subsequent measurement $d(\alpha + \Delta\alpha)$ is likely to produce a similar range value. This is not true of measurements coming from grass. This property is not self-evident from the marginal range distributions discussed in Section 3.2.2, because the order of measurements is lost when building such statistics. For example, in the case of a slanted

surface patch (large γ), range measurements have large marginal variance, and their distribution may be confused with that of a patch of grass if the sampling order is neglected.

The spatial coherence hypothesis, however, is weakened when the surface is partially hidden by vegetation, because laser beams have a certain probability of being intercepted by grass blades before hitting the surface. To minimize this effect, we can pre-process the measured data with a non-linear filter that outputs the maximum value of range within a window of contiguous samples. This filter exploits the fact that, facing a (partially hidden) surface, no measurement of points beyond such surface can be taken (see Fig. 12). Thus, if a laser beam within a small window hits the surface of an obstacle, then the maximum range measured within this window of samples is likely to correspond to an obstacle surface measurement. In a sense, the maximum-value filter acts as a "grass remover". This concept can be better formalized by looking at how the filter transforms the distribution of the input data. Let $\{d_1, d_2, \ldots, d_n\}$be a window of n equidistributed and mutually independent[6] range samples. Then, the cumulative distribution function (cdf) of $\hat{d} = \max\{d_1, d_2, \ldots, d_n\}$ is equal to:

$$\begin{aligned} F_{\hat{d}}(d) &= P(\hat{d} < d) = P(d_1 < d, d_2 < d, \ldots, d_n < d) \\ &= F_d(d)^n \end{aligned} \tag{8}$$

where $F_d(d)$ is the marginal cdf of the range measurements. By differentiation, we find the pdf of $\hat{d}$ to be

$$f_{\hat{d}}(d) = n F_d(d)^{n-1} f_d(d) \tag{9}$$

where $f_d(d)$ is the marginal pdf of the range measurements. Figure 14(b) shows the probability densities of filtered range measurements for the case of pure grass (solid line) and of a partially hidden surface (dashed line). It is seen that in the second case, the effect of the maximum-value filter is to concentrate mass around the values corresponding to the surface (thereby reducing variance), while, in the first case (pure grass), variance remains high.

Our surface detection algorithm works as follows. First, the data is subdivided into windows of n non-overlapping samples, and the maximum value $\hat{d}$ among the samples in each window is extracted. If n is large (in our case, greater than 3), the set of non-overlapping windows can be replaced by a n-sample sliding window. Then, the resulting sequence $\{\hat{d}\}$ is analyzed, to check for segments with high spatial coherence that provide evidence for surface patches. Spatial coherence can be expressed in terms of the differential characteristics of the data. In particular, we compute the magnitude of the second derivative of $\hat{d}$:

$$y = |\mathrm{d}^2\hat{d}/\mathrm{d}\alpha^2| \tag{10}$$

This has the advantage of producing a very small or null output value when the input is a linear ramp (corresponding to a locally flat surface, in any orientation). Thus, small values of y should indicate possible surface patches.

The identification of surface hits is not done on a scan-by-scan basis but on a sequence of scans. Let's assume that the analysis of a scan at time t indicates that a return from a laser beam fired in direction α, that yielded a range estimate $d(\alpha)$, is likely to have hit an obstacle surface. This hypothesis can be verified by observing the range estimate in directions adjacent to α at time $t + \Delta t$. Thus, hits are classified as surfaces only after they are compared against hits in previous scans. In essence, this methodology applies consistency of the scene through time, i.e., if Δt is sufficiently small, the relative positions of ladar and surface can change very little regardless of the relative motions and speeds of both ladar and surfaces. Thus, a hit in a direction with range estimate $d(\alpha)$ obtained at time t is likely to be an obstacle surface if there is a hit in direction $\alpha + \Delta\alpha$ with a range estimate $d(\alpha) + \Delta d$ at time $t + \Delta t$ for sufficiently small $\Delta\alpha$, Δd and Δt. This is the primary criterion for selecting candidate surface hits. Additional heuristic criteria can be added to remove false positives from such candidates. For example, candidate hits are removed when there are no other candidate hits nearby. We also remove candidate hits that lie outside the operative range [0.3 m 2.0 m]. As a reference, for our experiments $\Delta t = 0.2$ s is given by the 5 Hz rotation rate of the laser. Likewise, the value of Δd is determined by the maximum distance that the robot can move in Δt, i.e., given a maximum speed of 1 ms, the robot can move in $\Delta t = 0.2$ s by a distance of $\Delta d = 0.2$ m, that is, 10% of the operational range. Thus, the robot is able to observe an approaching object (either because of the motion of the object or the motion of the robot) up to 10 times before reaching it, which gives the algorithm time to evaluate the scene. Finally, in our setup we have $\Delta\alpha \approx 30^\circ$. There are many situations in which the relative change of orientation between the robot and obstacle in a Δt lapse is larger than $\Delta\alpha$ including a fast in-place

rotation of the robot, a fast straight motion of the robot passing by a stationary obstacle and any combination of these, among others (there are corresponding conditions when the robot is not moving but the obstacle is in motion, and when both are in motion). In these cases, the absence of a sufficiently small $\Delta\alpha$ does not mean that the obstacles cannot be detected but only that they cannot be tracked; when the value of $\Delta\alpha$ becomes sufficiently small, the obstacle will be detected, as a new different obstacle.

3.2.4. Experimental Results. Our algorithm was tested on scenes taken by the Acuity laser while mounted on the Urbie robot, as shown in Fig. 12. The experiments were run on eight scenes for periods of length between 15 and 90 seconds. The scenes included environments with sparse and dense grass, with and without obstacles (where the rocks were both trees and rocks), with obstacles in the clear and partially hidden and with the robot both stationary and on the move at various speeds. From the 1313 scans produced by these scenes we obtained 4 false alarms (i.e., four cases in which grass was misclassified as an obstacle surface); all the obstacles in the sequences were identified correctly, at some point of the run, i.e., not every obstacle is identified in every single frame in which it appears but every obstacle is identified as such, in at least one frame of the sequence. This classification is consistent with the hazard detection and avoidance objective of the task, in contrast to a target tracking task that would penalize the loss of the target during any frame of the sequence The right half of Fig. 16 shows the obstacle classification that corresponds to the scene in left half of Fig. 12, repeated in Fig. 16 as a reference. The lines shown in light gray indicate hits that are associated with obstacles. Note that both rocks (including the partially hidden one) have been detected correctly. Running times for the algorithm in a Pentium 4, 2.2 Ghz machine, are in the order of 450 μs and 215 μs per revolution (512 samples) for scenes with and without obstacles, respectively. Thus, the algorithm can be easily integrated into a real-time navigation system. Finally, the sequence of images in Fig. 17 shows some results of the obstacle detection and avoidance in tall grass using Urbie. In this case, the algorithm was ported to C++ and ran under VxWorks. At the time of the porting, the customized Acuity ladar of Urbie had been replaced by a Sick ladar with a sampling interval of 8.7 mradians (0.5 degrees), a scan span of 180° and an expected range error of 70 mm at 4 m. The sequence in Fig. 17 shows Urbie following a path that has an obstacle hidden by tall grass; the robot correctly detects

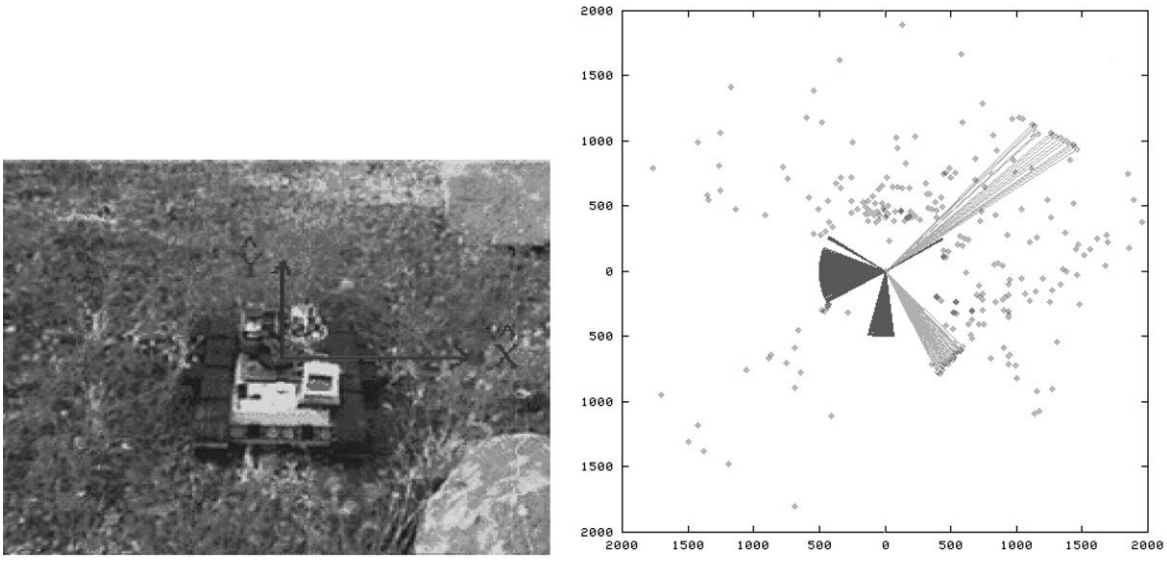

Figure 16. (a) The "Urbie" robot facing two rocks at 45° and 315° (repeated from Fig. 12 for reference). (b) Obstacle surface detection on the range data of Fig. 12.

Figure 17. Obstacle avoidance in tall grass using laser-based foliage classification.

the obstacle and steers to the left to avoid it. Images and movies of the results of these tests are available at telerobotics.jpl.nasa.gov/people/andres/{tmr,mars}.

Note that other techniques for discriminating vegetation from soil and obstacles using ladar data from a moving robot have been proposed (Hebert et al., 2002; Lacaze et al., 2002). These approaches, which are appropriate for use with a 2-axes ladar, are voxel-based representations of the 3-D data. This representation allows for the easy integration of measurements as the vehicle moves ahead (if accurate information about the vehicle's position and attitude is available). With respect to voxel-based techniques, our algorithm has a much lower computational complexity and does not require integration of data over multiple scans (although it does use multiple scans to rule out false positives), while providing excellent results with a single-axis ladar.

4. Conclusions

We have presented novel algorithms for obstacle detection and terrain classification for cross-country environments. Our techniques, which use data from a stereo color cameras and from a single-axis ladar, are at the core of JPL's perceptual system for autonomous off-road navigation and have been tested in the context of the several projects funded by the US Department of Defense. Our experimental result in the course of these projects have shown the viability and robustness of our algorithms for obstacle detection, color-based terrain typing and obstacle labeling, and ladar-based obstacle detection in grass.

Our more recent work is looking at further extending the terrain perception capabilities of a vehicle by using new sensors and new visual analysis algorithms. For example, color analysis can be made more robust by taking atmospheric effects into account (Nayar and Narasimhan, 2000). Texture features (Castano et al., 2001) could be used to either complement color classification or for terrain typing at night, when color cannot be used. Possible additional sensors include multispectral cameras in the thermal IR (Abedin et al., 2003), which also could be used for terrain classification at night, and polarization cameras, which may help detecting water bodies in the scene (Wolff, 1996).

Appendix

In this Appendix we compute the distribution of the range in the case of an obstacle partially hidden by grass. The geometry of the obstacle surface is represented in Fig. 13; we are interested in the distribution within the angular sector [0, A]. To simplify our treatment, we will henceforth assume that $K = \cot\gamma < 0$, as in Fig. 13 (the case $K > 0$ is derived straightforwardly). It is assumed that the area between the ladar and the obstacle is uniformly covered by grass with spatial density $\bar{\lambda}$ and constant blade diameter h (then $\lambda = h\bar{\lambda}$, as defined in Section 3.2.2).

We first compute the cumulative distribution function (cdf) of the range d:

$$F_d(d) = \int_{\alpha=0}^{\mathrm{A}} P(d(\alpha) < d) f_\alpha(\alpha)\, \mathrm{d}\alpha = \frac{1}{\mathrm{A}} \int_{\alpha=0}^{\mathrm{A}} P(d(\alpha) < d)\, \mathrm{d}\alpha \qquad \text{(A.1)}$$

where we used the total probability theorem, conditioning with respect to the measuring angle α, and assuming that the pdf $f_\alpha(\alpha)$ of α is uniform within [0, A]. Using

approximation (7), and remembering that $K < 0$, we can write

$$P(d(\alpha) < d) = \begin{cases} 0, & d < 0 \\ 1 - e^{-\lambda d}, & 0 \le d \le d(\alpha) \\ 1, & d > d(\alpha) \end{cases}$$
$$= \begin{cases} 0, & d < 0 \\ 1 - e^{-\lambda d}, & d \ge 0\,,\ \alpha \ge \dfrac{D-d}{DK} \\ 1, & \alpha < \dfrac{D-d}{DK} \end{cases} \tag{A.2}$$

Combining (A.1) with (A.2), one obtains the cdf of the range within the sector of angle A:

$$F_d(d) = \begin{cases} 0, & d < 0 \\ 1 - e^{-\lambda d}, & 0 \le d < D \\ \dfrac{1}{A}\left(\displaystyle\int_0^{\frac{D-d}{DK}} \mathrm{d}\alpha + \int_{\frac{D-d}{DK}}^{A} (1 - e^{-\lambda d})\,\mathrm{d}\alpha \right), & D \le d < D(1 - AK) \\ 1, & d \ge D(1 - AK) \end{cases}$$
$$= \begin{cases} 0, & d < 0 \\ 1 - e^{-\lambda d}, & 0 \le d < D \\ e^{-\lambda d}\left(\dfrac{D-d}{ADK} - 1 \right) + 1\,, & D \le d < D(1 - AK) \\ 1, & d \ge D(1 - AK) \end{cases} \tag{A.3}$$

and, by differentiation, its pdf:

$$f_d(d) = \begin{cases} 0, & d < 0 \\ \lambda e^{-\lambda d}, & 0 \le d < D \\ \lambda e^{-\lambda d}\left(1 - \dfrac{D - d + 1/\lambda}{ADK} \right), & D \le d < D(1 - AK) \\ 0, & d \ge D(1 - AK) \end{cases} \tag{A.4}$$

This pdf is represented in Fig. 14 (left, dashed line) for $\lambda = 1/\mathrm{m}$, $\mathrm{A} = 6^\circ$, $\gamma = 107^\circ$ (and therefore $K = \cot\gamma = -3$), and $D = 1.5$ m.s

Acknowledgments

This research in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. This work was supported by DARPA, Mobile Autonomous Robot Software Program, under contract NAS7-1407; DARPA, Tactical Mobile Robotics Program, under contract NAS-1407; and the Army Research Laboratory, DEMO III Experimental Unmanned Vehicle Program.

Notes

1. A similar OD algorithm was used with ladar measurements in Hong et al. (2000).
2. Since there is a one-to-one correspondence between visible 3-D points $\boldsymbol{p}$ and their projections $\bar{\boldsymbol{p}}$ onto the image plane, we will say that two pixels $\bar{\boldsymbol{p}}_1$ and $\bar{\boldsymbol{p}}_2$ are compatible when the corresponding 3-D points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are.
3. This approximation is acceptable when α is not too small.
4. The curves in Figure 8 were built using data from the ASTER Spectral Library (Hook, http://speclib.jpl.nasa.gov).
5. We use a simple graphical user interface that allows one to quickly hand-label images in a training data set.
6. The original color images can be obtained by sending email to manduchi@soe.ucsc.edu.
7. Statistical independence is an acceptable hypothesis given our assumption that no two laser beams can hit the same grass blade, and assuming dense grass.

References

Abedin, M.N. et al. 2003. Multicolor focal plan array detector technology: A review. In *SPIE Int. Symp. on Optical Science and Technology*, San Diego.

Adams, M. 2000. Lidar design, use, and calibration concepts for correct environmental detection. *IEEE Trans. Robotics Automat.*, 16:753–761.

Badal, S., Ravela, S., Draper, B., and Hanson, A. 1994. A practical obstacle detection and avoidance system. In *2nd IEEE Workshop on Application of Computer Vision*.

Batavia, P. and Singh, S. 2002. Obstacle detection in smooth high curvature terrain. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA '02)*.

Baten, S., Mandelbaum, R., Luetzeler, M., Burt, P., and Dickmanns, E. 1998. Techniques for autonomous, off-road navigation. *IEEE Intelligent Systems Magazine*, 57–65.

Bellutta, P., Manduchi, R., Matthies, L., Owens, K., and Rankin, A. 2000. Terrain perception for Demo III". *Intelligent Vehicles Conference*.

Bergh, C., Kennedy, B., Matthies, L., and Johnson, A. 2000. A compact and low power two-axis scanning laser rangefinder for mobile robots. In *Seventh Mechatronics Forum International Conference*, Atlanta, Georgia.

Bishop, C.M. 1995. *Neural Networks for Pattern Recognition*. Oxford University Press.

Broggi, A., Bertozzi, M., Fascioli, A., Guarino LoBianco, C., and Piazzi, A. 2000. Visual perception of obstacles and vehicles for platooning. *IEEE Trans. Intell. Transport. Sys.*, 1(3).

Buluswar, S.D. and Draper, B.A. 2002. Color models for outdoor machine vision. *Computer Vision and Image Understanding*, 85(2):71–99.

Castano, A., and Matthies, L. 2003. Foliage discrimination using a rotating Ladar. In *Proc. IEEE Intl. Conf. Robotics and Automation*, Taipei, Taiwan.

Castano, R., Manduchi, R., and Fox, J. 2001.Classification experiments on real-world textures. *Workshop on Empirical Evaluation in Computer Vision*, Kauai, HI.

Elachi, C. 1987. *Introduction to the Physics and Techniques of Remote Sensing*. John Wiley and Sons,

Healey, G. 1992. Segmenting images using normalized color. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(1):64–73.

Hebert, M., Vandapel, N., Keller, S., and Donamukkala, R.R. 2002. Evaluation and comparison of terrain classification techniques from LADAR data for autonomous navigation. *Army Science Conference*, Orlando.

Hong, T., Abrams, M., Chang, T., and Shneier, M.O. 2000. An intelligent world model for autonomous off-road driving. *Computer Vision and Image Understanding*,

Hook, S. *ASTER Spectral Library*. http://speclib.jpl.nasa.gov.

Horn, B.K.P. 1986. *Robot Vision*. MIT Press: Cambridge, Massachusetts.

Huang, J., Lee, A.B., and Mumford, D. 2000. Statistics of range images. In *IEEE Conf. Computer Vision and Pattern Recognition*, Hilton Head.

Lacaze, A., Moscovitz, Y., DeClaris, N., and Murphy, K. 1998. Path planning for autonomous vehicles driving over rough terrain. In *IEEE ISIC/CIRA/ISAS Joint Conference*.

Lacaze, A., Murphy, K., and DelGiorno, M. 2002. Autonomous mobility for the DEMO III Experimental Unmanned Vehicle. *Association for Unmanned Vehicle Systems—Unmanned Vehicle*.

Leung T. and Malik, J. 1997. On perpendicular texture or: Why do we see more flowers in the distance? In *IEEE Conf. Computer Vision and Pattern Recognition*, San Juan, Puerto Rico.

Macedo, J., Manduchi, R., and Matthies, L. 2000. Ladar-based discrimination of grass from obstacles for autonomous navigation. In *Proc. Intl. Symposium on Experimental Robotics*, pp.111–120,

Maloney, L.T. and Wandell, B.A. 1986. Color constancy: A method for recovering surface spectral reflectance. *J. Opt. Soc. Amer. A*, 3:29–33.

Manduchi, R. 2004. Outdoor color classification using only one training image. In *European Conference on Computer Vision 2004*, Prague.

Matthies, L., Xiong, Y., Hogg, R., Zhu, D., Rankin, A., and Kennedy, B. 2000. A portable, autonomous, urban reconnaissance robot. In *International Conference on Intelligent Autonomous Systems*, Venice, Italy.

Matthies, L. and Grandjean, P. 1994. Stochastic performance modeling and evaluation of obstacle detectability with imaging range sensors. *IEEE Transactions on Robotics and Automation, Special Issue on Perception-based Real World Navigation*, 10(6).

Matthies, L., Kelly, A., Litwin, T., and Tharp, G. 1996. Obstacle detection for unmanned ground vehicles: a progress report. *Robotics Research 7*, Springer-Verlag.

Matthies, L., Litwin, T., Owens, K., Rankin, A., Murphy, K., Coombs, D., Gilsinn, J., Hong, T., Legowik, S., Nashman, M., and Yoshimi, B. 1998. Performance evaluation of UGV obstacle detection with CCD/FLIR stereo vision and LADAR. In *IEEE ISIC/CIRA/ISAS Joint Conference* .

McCartney, E.J. 1976. *Optics of the Atmosphere: Scattering by Molecules and Particles*. John Wiley and Sons: New York.

McLachlan, G. and Peel, D. 2000. *Finite Mixture Models*. John Wiley and Sons.

Mehlhorn, K. 1984. *Data Structures and Efficient Algorithms*. Springer Verlag.

Nayar, S.K., and Narasimhan, S.G. 2000. Vision in bad weather. In *IEEE Conf. Computer Vision and Pattern Recognition*.

Rankin, A., Owens, K., Matthies, L., and Litwin, T. 1998. Terrain-adaptive gaze and velocity control for ugv obstacle detection. In *Association for Unmanned Vehicle Systems International Annual Symposium*.

Ripley, B. 1996. *Pattern Recognition and Neural Networks*. Cambridge University Press.

Roberts, D., Smith, M., and Adams, J. 1993. Green vegetation, nonphotosynthetic vegetation, and soils in AVIRIS data. *Remote Sens. Environ*., 44:255–269.

Shoemaker, C.M. and Bornstein, J.A. 1998. The Demo III UGV program: A testbed for autonomous navigation Research. In *IEEE International Symposium on Intelligent Control*, Gaithersburg, MD.

Shi, X. and Manduchi, R. 2003. A study on Bayes feature fusion for image classification. In *IEEE Workshop on Statistical Analysis in Computer Vision*, Madison, WI.

Singh, S., and Keller, P. 1991. Obstacle detection for high speed autonomous navigation. In *IEEE Int. Conference on Robotics and Automation*, Sacramento, CA, pp. 2798–2805.

Talukder, A., Manduchi, R., Castano, R., Owens, K., Matthies, L., Castano, A., and Hogg, R. 2002. Autonomous terrain characterization and modeling for dynamic control of unmanned vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland.

Talukder, A., Manduchi, R., Rankin, A., and Matthies, L. 2002. Fast and reliable obstacle detection and segmentation for cross-country navigation. In *IEEE Intelligent Vehicles Symposium*, Versailles, France.

Williamson, T. and Thorpe, C. 1998. A specialized multibaseline stereo technique for obstacle detection. In *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 238–244.

Wolff, L. 1996. Polarization phased-based method for material classification and object recognition in computer vision. In *IEEE Conf. Computer Vision and Pattern Recognition*.

Zhang, Z., Weiss, R., and Hanson, A.R. 1994. Qualitative obstacle detection. In *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 554–559.

http://www.gao.gov/new.items/d01311.pdf

http://www.grandchallenge.org

Roberto Manduchi is an Assistant Professor of Computer Engineering at the University of California, Santa Cruz. Before joining UCSC, he was a Senior Staff Member at the Jet Propulsion Laboratory, an Adjunct Assistant Professor at the University of Southern California, and a Senior Engineer at Apple Computer, Inc. He received a "Dottorato di Ricerca" from the University of Padova, Italy, in 1993. He works in the area of computer vision and sensor processing, with applications to robotics, communications, and assistive technology for the blind. His research has bee funded by NSF, DARPA, and NASA.

Andres Castano received the B.S. and M.E.E. degrees from the University of Los Andes, Bogota, Colombia, and the M.S. and Ph.D. degrees from the University of Illinois at Urbana-Champaign, all in Electrical Engineering. At Urbana-Champaign he worked at the Beckman Institute in visual servo-control of manipulators and panoramic cameras with very large depth-of-field. Later he joined the Information Sciences Institute at the University of Southern California where he was the lead roboticist of the Conro project on reconfigurable robots. Currently he is with the NASA/Jet Propulsion Laboratory at the California Institute of Technology. His research interests are robotics, computer vision and graph theory.

Ashit Talukder obtained a Ph.D. degree in ECE from Carnegie Mellon University (CMU). He is currently a senior researcher at Jet Propulsion Laboratory. He has extensive experience in signal and image processing, computer vision, machine learning, pattern recognition, controls and optimization. His current research interests are in robotics, intelligent sensor networks, remote sensing, biomedical signal processing, and product inspection. He has led several DARPA, NIH, and NASA sponsored projects. He has a provisional patent for the design of software architecture and algorithms for biomedical data analysis and visualization, which was nominated for the NASA software of the year. He has over 40 publications, has chaired several SPIE conference sessions & given 3 invited conference talks, and is on the technical program committee of the Optical Pattern Recognition Conference in the SPIE Defense and Security Symposium. He is a reviewer on IEEE Trans Image Processing, Neural Networks, Optical Engineering, and Neurocomputing journals.

Larry Matthies received his Ph.D. in Computer Science from Carnegie Mellon University (CMU) in 1989 and has been at the Jet Propulsion Laboratory since then, where he is now supervisor of the Machine Vision Group. His research interests are in perception for autonomous navigation of robotic ground and air vehicles. He developed the first successful real-time stereo vision system for autonomous, cross-country obstacle avoidance by mobile robots and demonstrated the first accurate, stereo vision-based visual odometry algorithm. Both of these algorithms were used in the 2003/2004 Mars Exploration Rover (MER) mission. His group developed the Descent Image Motion Estimation System (DIMES), which used feature tracking in down-looking imagery to estimate the horizontal velocity of the MER landers during terminal descent. He also conducts research on multi-sensor terrain classification and scene motion analysis for mobile robots for Earth-based applications. He is an Adjunct Professor in the Computer Science Department at the University of Southern California and is a member of the editorial board for the journal "Autonomous Robots".