Proceedings of the AAAI National Conference on Artificial Intelligence, Austin, TX, July/August 2000

Appearance-Based Obstacle Detection with Monocular Color Vision

Iwan Ulrich and Illah Nourbakhsh

The Robotics Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
iwan@ri.cmu.edu, illah@ri.cmu.edu

Abstract

This paper presents a new vision-based obstacle detection method for mobile robots. Each individual image pixel is classified as belonging either to an obstacle or the ground based on its color appearance. The method uses a single passive color camera, performs in real-time, and provides a binary obstacle image at high resolution. The system is easily trained by simply driving the robot through its environment. In the adaptive mode, the system keeps learning the appearance of the ground during operation. The system has been tested successfully in a variety of environments, indoors as well as outdoors.

1. Introduction

Obstacle detection is an important task for many mobile robot applications. Most mobile robots rely on range data for obstacle detection. Popular sensors for *range-based* obstacle detection systems include ultrasonic sensors, laser rangefinders, radar, stereo vision, optical flow, and depth from focus. Because these sensors measure the distances from obstacles to the robot, they are inherently suited for the tasks of obstacle detection and obstacle avoidance. However, none of these sensors is perfect. Ultrasonic sensors are cheap but suffer from specular reflections and usually from poor angular resolution. Laser rangefinders and radar provide better resolution but are more complex and more expensive. Most depth from X vision systems require a textured environment to perform properly. Moreover, stereo vision and optical flow are computationally expensive.

In addition to their individual shortcomings, all range-based obstacle detection systems have difficulty detecting small or flat objects on the ground. Reliable detection of these objects requires high measurement accuracy and thus precise calibration. Range sensors are also unable to distinguish between different types of ground surfaces. This is a problem especially outdoors, where range sensors are usually unable to differentiate between the sidewalk pavement and adjacent flat grassy areas.

While small objects and different types of ground are difficult to detect with range sensors, they can in many cases be easily detected with color vision. For this reason, we have developed a new *appearance-based* obstacle detection system that is based on passive monocular color vision. The heart of our algorithm consists of detecting pixels different in appearance than the ground and classifying them as obstacles. The algorithm performs in real-time, provides a high-resolution obstacle image, and operates in a variety of environments. The algorithm is also very easy to train.

The fundamental difference between range-based and appearance-based obstacle detection systems is the obstacle criterion. In range-based systems, obstacles are objects that protrude a minimum distance from the ground. In appearance-based systems, obstacles are objects that differ in appearance from the ground.

2. Related Work

While an extensive body of work exists for range-based obstacle detection, little work has been done in appearance-based obstacle detection (Everett 1995). Interestingly, Shakey, the first autonomous mobile robot, used a simple form of appearance-based obstacle detection (Nilsson 1984). Because Shakey operated on textureless floor tiles, obstacles were easily detected by applying an edge detector to the monochrome input image. However, Shakey's environment was artificial. Obstacles had non-specular surfaces and were uniformly coated with carefully selected colors. In addition, the lighting, walls, and floor were carefully set up to eliminate shadows.

Horswill used a similar method for his mobile robots Polly and Frankie, which operated in a real environment (Horswill 1994). Polly's task was to give simple tours of the 7th floor of the MIT AI lab, which had a textureless carpeted floor. Obstacles could thus also be detected by applying an edge detector to the monochrome input images, which were first subsampled to 64×48 pixels and then smoothed with a 3×3 low-pass filter.

Shakey and Polly's obstacle detection systems perform well as long as the background texture constraint is satisfied, i.e., the floor has no texture and the environment is uniformly illuminated. False positives arise if there are shiny floors, boundaries between carpets, or shadows. False negatives arise if there are weak boundaries between the floor and obstacles.

Turk and Marra developed an algorithm that uses color instead of edges to detect obstacles on roads with minimal texture (Turk and Marra 1986). Similar to a simple motion detector, their algorithm detects obstacles by subtracting two consecutive color images from each other. If the ground has substantial texture, this method suffers from similar problems as systems that are based on edge detection. In addition, this algorithm requires either the robot or the obstacles to be in motion.

While the previously described systems fail if the ground is textured, stereo vision and optical flow systems actually require texture to work properly. A thorough overview of such systems is given by Lourakis and Orphanoudakis (Lourakis and Orphanoudakis 1997). They themselves developed an elegant method that is based on the registration of the ground between consecutive views of the environment, which leaves objects extending from the ground unregistered. Subtracting the reference image from the warped one then determines protruding objects without explicitly recovering the 3D structure of the viewed scene. However, the registration step of this method still requires the ground to be textured.

In her master's thesis, Lorigo extended Horswill's work to domains with texture (Lorigo, Brooks, and Grimson 1997). To accomplish this, her system uses color information in addition to edge information. The key assumption of Lorigo's algorithm is that there are no obstacles right in front of the robot. Thus, the ten bottom rows of the input image are used as a reference area. Obstacles are then detected in the rest of the image by comparing the histograms of small window areas to the reference area. The use of the reference area makes the system very adaptive. However, this approach requires the reference area to always be free of obstacles. To minimize the risk of violating this constraint, the reference area can not be deep. Unfortunately, a shallow reference area is not always sufficiently representative for pixels higher up in the image, which are observed at a different angle. A particular problem are highlights, which usually occur higher up in an image. The method performs in real-time, but uses an image resolution of only 64×64 pixels.

Similar to Lorigo's method, our obstacle detection algorithm also uses color information and can thus be used in a wide variety of environments. Unlike the previously described systems that are all purely reactive, our method permits the use of a deeper reference area by learning the appearance of the ground over several observations. In addition, the learned data can easily be stored and later be reused. Like Lorigo's method, our system also uses histograms and a reference area ahead of the robot, but does not impose the constraint that the reference area is always free of obstacles. In addition, our methods provides binary obstacle images at high resolution in real-time.

3. Appearance-Based Obstacle Detection

Our obstacle detection system is purely based on the appearance of individual pixels. Any pixel that differs in appearance from the ground is classified as an obstacle. The method is based on three assumptions that are reasonable for a variety of indoor and outdoor environments:

1. Obstacles differ in appearance from the ground.
2. The ground is relatively flat.
3. There are no overhanging obstacles.

The first assumption allows us to distinguish obstacles from the ground, while the second and third assumptions allow us to estimate the distances between detected obstacles and the camera.

The classification of a pixel as representing an obstacle or the ground can be based on a number of local visual attributes, such as intensity, color, edges, and texture. It is important that the selected attributes provide information that is rich enough so that the system performs reliably in a variety of environments. The selected attributes should also require little computation time so that real-time performance can be achieved without dedicated hardware. The less computationally expensive the attribute, the higher the obstacle detection update rate, and the faster a mobile robot can travel safely.

To best satisfy these requirements, we decided to use color information as our primary cue. Color has many appealing attributes, although little work has lately been done in color vision for mobile robots. Color provides more information than intensity alone. Compared to texture, color is a more local attribute and can thus be calculated much faster. Systems that solely rely on edge information can only be used in environments with textureless floors, as in the environments of Shaky and Polly. Such systems also have more difficulty differentiating between shadows and obstacles than color-based systems.

For many applications, it is important to estimate the distance from the camera to a pixel that is classified as an obstacle. With monocular vision, a common approach to distance estimation is to assume that the ground is relatively flat and that there are no overhanging obstacles. If these two assumptions are valid, then the distance is a monotonically increasing function of the pixel height in the image. The estimated distance is correct for all obstacles at their base, but the higher an obstacle part is above the ground, the more the distance is overestimated. The simplest approach of dealing with this problem consists of only using the obstacle pixels that are lowest for each column of the image. A more sophisticated approach consists of grouping obstacle pixels and assigning the shortest distance to the entire group.

4. Basic Approach

Our basic approach can best be explained with an example and a *simplified* version of our method. Figure 1 shows a color input image with three reference areas of different depths on the left and the corresponding outputs of the simplified version on the right. The remainder of this section describes the details of the simplified version. Unlike the full version of our method, the simplified version has no memory and uses only one input image. However, all functions of the simplified version are also used by the full version of our method, which will be described in detail in section five.

The simplified version of our appearance-based obstacle detection method consists of the following four steps:

1. Filter color input image.
2. Transformation into HSI color space.
3. Histogramming of reference area.
4. Comparison with reference histograms.

In the first step, the 320×260 color input image is filtered with a 5×5 Gaussian filter to reduce the level of noise.

In the second step, the filtered RGB values are transformed into the HSI (hue, saturation, and intensity) color space. Because color information is very noisy at low intensity, we only assign valid values to hue and saturation if the corresponding intensity is above a minimum value. Similarly, because hue is meaningless at low saturation, hue is only assigned a valid value if the corresponding saturation is above another minimum value. An appealing attribute of the HSI model is that it separates the color information into an intensity and a color component. As a result, the hue and saturation bands are less sensitive to illumination changes than the intensity band.

In the third step, a trapezoidal area in front of the mobile robot is used for reference. The valid hue and intensity values of the pixels inside the trapezoidal reference area are histogrammed into two one-dimensional histograms, one for hue and one for intensity. The two histograms are then low-pass filtered with a simple average filter. Histograms are well suited for this application, as they naturally represent multi-modal distributions. In addition, histograms require very little memory and can be computed in little time.

In the fourth step, all pixels of the filtered input image are compared to the hue and the intensity histograms. A pixel is classified as an obstacle if either of the two following conditions is satisfied:

i) The hue histogram bin value at the pixel's hue value is below the hue threshold.
ii) The intensity histogram bin value at the pixel's intensity value is below the intensity threshold.

If none of these conditions are true, then the pixel is classified as belonging to the ground. In the current implementation, the hue and the intensity thresholds are set to 60 and 80 pixels respectively. The hue threshold is chosen smaller than the intensity threshold because not every pixel is assigned a valid hue value.

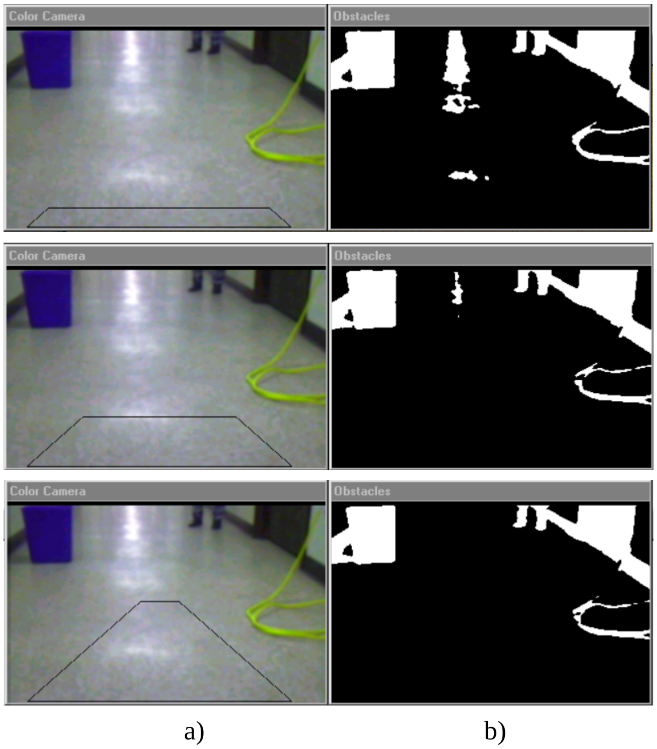

Figure 1: a) Input color image with trapezoidal reference area
b) Binary obstacle output image

As shown in Figure 1, the simplified version of our algorithm performs quite well. Independent of the depths of the reference area, the method detects the lower parts of the right and left corridor walls, the trash can on the left, the door on the right, and the shoes and pants of the person standing a few meters ahead. In particular, the algorithm also detects the cable lying on the floor, which is very difficult to detect with a range-based sensor. The three example images also demonstrate that the deeper the reference area, the more representative the area is for the rest of the image. In the case of the shallow reference area of the top image, the method incorrectly classifies a highlight as an obstacle that is very close to the robot.

5. Implementation

In the previous section, we have shown how the reference area in front of the mobile robot can be used to detect obstacles in the rest of the image. Obviously, this approach only works correctly if no obstacles are present inside the trapezoidal reference area. Lorigo's method, which assumes that the area immediately in front of the robot is free of obstacles, is thus forced to use a reference area with a shallow depth to reduce the risk of violating this constraint. However, as demonstrated in Figure 1, a shallow reference area is not always sufficiently representative.

In order to use a deeper reference area, we avoid the strong assumption that the area immediately ahead of the robot is free of obstacles. Instead, we assume that the ground area over which the mobile robot traveled was free of obstacles. In our current implementation, the reference area is about one meter deep. Therefore, whenever the mobile robot travels relatively straight for more than one meter, we can assume that the reference area that was captured one meter ago was free of obstacles. Conversely, if the robot turns a substantial amount during this short trajectory, it is no longer safe to assume that the captured reference areas were free of obstacles.

The software implementation of the algorithm uses two queues: a *candidate queue* and a *reference queue*. For each acquired image, the hue and intensity histograms of the reference area are computed as described in the first three steps of section four. These histograms together with the current odometry information are then stored in the candidate queue.

At each sample time, the current odometry information is compared with the odometry information of the items in the candidate queue. In the first pass, items whose orientation differs by more than 18° from the current orientation are eliminated from the candidate queue. In the second pass, items whose positions differ by more than 1 m from the current position are moved from the candidate queue into the reference queue.

The histograms of the reference queue are then combined using a simple OR function, resulting in the *combined hue histogram* and the *combined intensity histogram*. The pixels in the current image are then compared to the combined hue and the combined intensity histogram as described in the fourth step of section four.

To train the system, one simply leads the robot through the environment, avoiding obstacles manually. This allows us to easily train the mobile robot in a new environment. The training result consists of the final combined hue and combined intensity histogram, which can easily be stored for later usage as they require very little memory. After training, these histograms are used to detect obstacles as described in the fourth step of section four.

6. Operation Modes

We have currently implemented three operation modes: *regular*, *adaptive,* and *assistive*. In the regular mode, the obstacle detection system relies on the combined histograms learned during training. While this works well in many indoor environments with static illumination, the output contains several false positives if the lighting conditions are different than during training.

In the adaptive mode, the system is capable of adapting to changes in illumination. Unlike the regular version, the adaptive version learns while it operates. The adaptive version uses two sets of combined histograms: a *static* set and a *dynamic* set. The static set simply consists of the combined histograms learned during the regular manual training. The dynamic set is updated during operation as described in section five as if the robot was being manually trained. The two sets are then combined with the OR function. With this approach, we would only add learned items to the reference queue, but never eliminate an item from it. This approach would correspond to the algorithm never forgetting a learned item. For the adaptive set, we actually chose to limit its memory, so that it forgets items that it learned a long distance ago by eliminating these items from the dynamic reference queue. In our current implementation, the dynamic reference queue only retains the ten most recent items.

The assistive mode is well suited for applications of teleoperated and assistive devices. This mode uses only the dynamic set. To start, one simply drives the robot straight ahead for a few meters. During the short training, the reference queue is quickly filled with valid histograms. When the obstacle detection module detects enough ground pixels, the obstacle avoidance module takes control of the robot, freeing the user from the task of obstacle avoidance. If the robot arrives at a different surface, the robot might stop because the second surface is incorrectly classified as an obstacle. The user can then manually override the obstacle avoidance module by driving the robot 1-2 meters forward over the new surface. During this short user intervention, the adaptive obstacle detection system learns the new surface. When it detects enough ground pixels, the obstacle avoidance module takes over again.

7. Experimental Results

For experimental verification, we implemented our algorithm on a computer-controlled electric wheelchair, which is a product of KIPR from Oklahoma. We added quadrature encoders to the wheelchair to obtain odometry information.

The vision software runs on a laptop computer with a Pentium II processor clocked at 333 MHz. The laptop is connected to the wheelchair's 68332 microcontroller with a serial link. Images are acquired from a Hitachi KPD-50 color CCD camera, which is equipped with an auto-iris lens. The camera is connected to the laptop with a PCMCIA framegrabber from MRT. The entire system is shown in Figure 2.

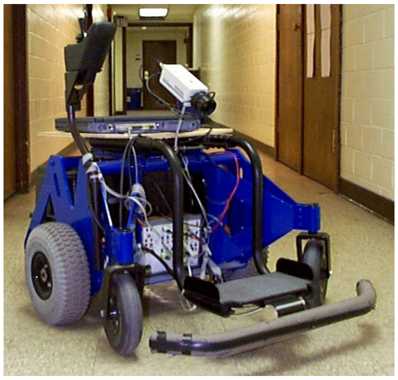

Figure 2: Experimental platform.

Figure 3 shows seven examples of our obstacle detection algorithm. The left images show the original color input images, while the right images show the corresponding binary output images. It is important to note that no additional image processing like blob filtering was applied to the output images. Color versions of the images, as well as additional examples, are available at http://www.cs.cmu.edu/~iwan/abod.html.

The first five images were taken indoors at the Carnegie Museum of Natural History in Pittsburgh. Figure 3a shows one of the best results that we obtained. The ground is almost perfectly segmented from obstacles, even for objects that are more than ten meters ahead. Figure 3b includes a moving person in the input image. Although some parts of the shoes are not classified as obstacles, enough pixels are correctly classified for a reliable obstacle detection. The person leaning onto the left wall is also classified as an obstacle. In Figure 3c, a thin pillar is detected as an obstacle, while its shadow is correctly ignored. However, the system incorrectly classifies a shadow that is cast from an adjacent room on the left as an obstacle. In Figure 3d, the lighting conditions are very difficult, as outdoor light illuminates a large part of the floor through a yellow window. Nevertheless, the algorithm performs well and correctly labels the yellowish pixels as floor. Figure 3e shows an example where our algorithm fails to detect an obstacle. The color of the yellow leg in the center is too similar to the floor. The algorithm also has trouble detecting the lower part of the left wall, which is made of the same material as the floor. However, the darker pixels at the bottom of the wall and the leg are detected correctly due to the high resolution of the image.

The last two images were taken outdoors on a sidewalk. In Figure 3f, both sides of the sidewalk, the bushes and the cars, are detected as obstacles. Moreover, the two large shadows are perfectly ignored. In Figure 3g, the sidewalk's right border is painted yellow. The algorithm reliably detects the grass on the left side and the yellow marking on the right side. Although the yellow marking is not really an obstacle, it does indicate the sidewalk border so that its detection is desirable in most cases. This example is also an argument for the use of color, as the yellow marking would not be recognized as easily with a monochrome camera.

With the current hardware and software implementation, the execution time for the processing of an image of 320 × 260 pixels is about 200 ms. It is important to note that little effort was spent on optimizing execution speed. Beside code optimization, faster execution speeds could also be achieved by subsampling the input image. Another possibility would be to only apply the algorithm to regions of interest, e.g., the lower portion of the image. By only using the bottom half of the image, which still corresponds to a look-ahead of two meters, the current system would achieve an update rate of 10 Hz, which is similar to a fast sonar system. However, our vision system provides information at a much higher resolution than is possible with sonars.

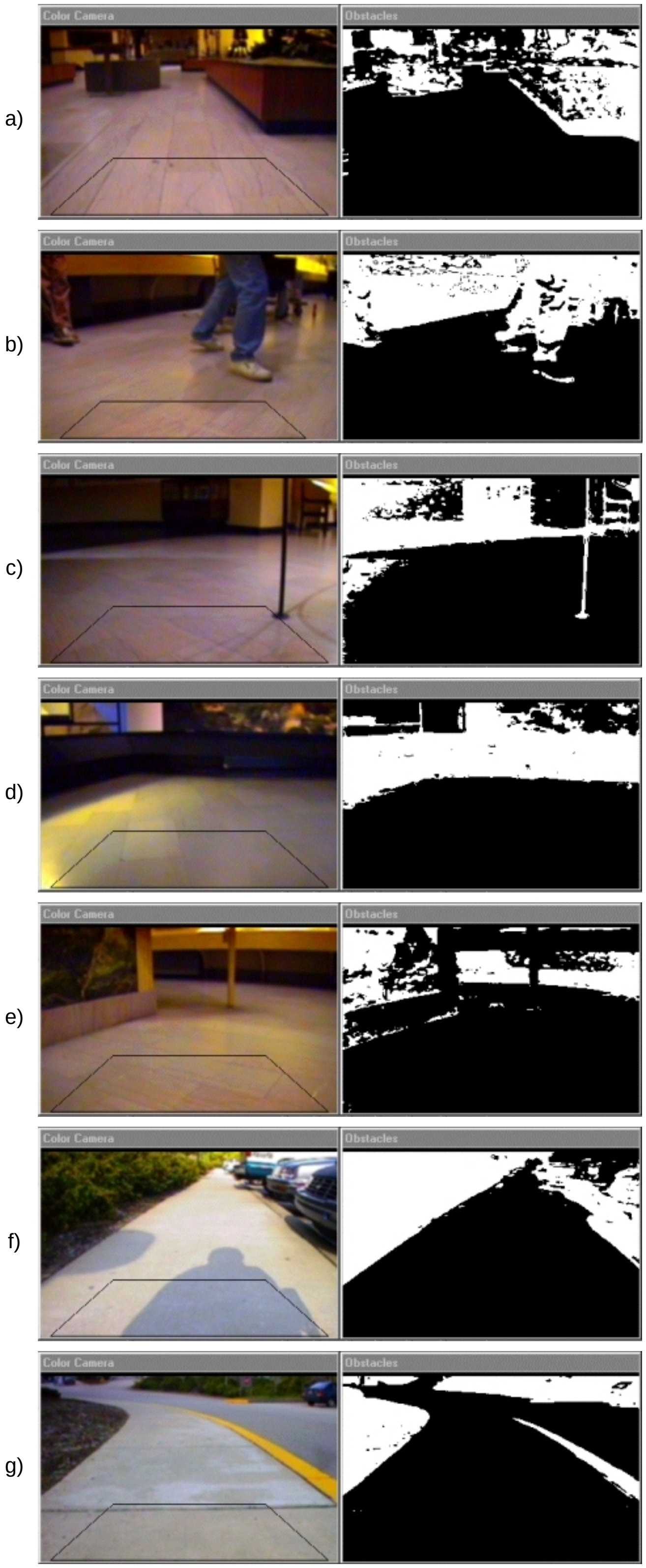

Figure 3: Experimental results.

To further test the performance of our obstacle detection system, we implemented a very simple reflexive obstacle avoidance algorithm on the electric wheelchair. The algorithm is similar to the one used by Horswill's robots (Horswill 1994), but uses five columns instead of three. Not surprisingly, after driving the robot for a few meters in the assistive operation mode, the obstacle detection output allows the wheelchair to easily follow sidewalks and corridors, and avoid obstacles. Another research group at the Robotics Institute has successfully combined our method with stereo vision to provide obstacle detection for an all terrain vehicle (Soto et al. 1999).

8. Evaluation

Our system works well as long as the three assumptions about the environment stated in section three are not violated. We have experienced only a few cases where obstacles violate the first assumption by having an appearance similar to the ground. Figure 3e shows such a case. Including additional visual clues might decrease the rate of false negatives to an even lower number.

We have never experienced a problem with the second assumption that the ground is flat, because our robots are not intended to operate in rough terrain. A small inclination of a sidewalk introduces small errors in the distance estimate. This is not really a problem as the errors are small for obstacles that are close to the robot.

Our system overestimates the distance to overhanging obstacles, which violate the third assumption. This distance estimate error can easily lead to collisions, even when the pixels are correctly classified as obstacles. Although truly overhanging obstacles are rare, tabletops are quite common and have the same effect. However, it is important to note that tabletops also present a problem for many mobile robots that are equipped with range-based sensors, as tabletops are often outside their field of view. The simplest way to detect tabletops is probably to combine our system with a range-based sensor. For example, adding one or two low-cost wide-angle ultrasonic sensors for the detection of tabletops would make our system much more reliable in office environments.

9. Further Improvements

The current algorithm could be improved in many ways. It will be interesting to investigate how color spaces other than HSI perform for this application. Examples of promising color spaces are YUV, normalized RGB, and opponent colors. The final system could easily combine bands from several color spaces using the same approach. Several bands could also be combined by using higher-dimensional histograms, e.g., a two-dimensional histogram for hue and saturation. In addition, several texture measures could be implemented as well.

For large environments with different kinds of surfaces, it would be preferable to have a set of learned histograms for each type of surface. Combining our obstacle detection method with a localization method would allow us to benefit from room-specific histogram sets. In particular, we recently developed a topological localization method that is a promising candidate for combination, because its vision-based place recognition module also relies heavily on the appearance of the ground (Ulrich and Nourbakhsh 2000).

Another possible improvement consists of enlarging the field of view by using a wide-angle lens or a panoramic camera system like the Omnicam. However, it is unclear at this time what the consequences of the resulting reduced resolution will be.

Another promising approach consists of combining the current appearance-based sensor system with a range-based sensor system. Such a combination is particularly appealing due to the complementary nature of the two systems.

10. Conclusion

This paper presented a new method for obstacle detection with a single color camera. The method performs in real-time and provides a binary obstacle image at high resolution. The system can easily be trained and has performed well in a variety of environments, indoors as well as outdoors.

References

Everett, H.R. 1995. *Sensors for Mobile Robots: Theory and Applications*. A K Peters, Wellesley, Massachusetts.

Horswill, I. 1994. Visual Collision Avoidance by Segmentation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 902-909.

Lorigo, L.M.; Brooks, R.A.; and Grimson, W.E.L. 1997. Visually-Guided Obstacle Avoidance in Unstructured Environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 373-379.

Lourakis, M.I.A., and Orphanoudakis, S.C. 1997. Visual Detection of Obstacles Assuming a Locally Planar Ground. Technical Report, FORTH-ICS, TR-207.

Nilsson, N.J. 1984. Shakey the Robot. Technical Note 323, SRI International.

Soto, A.; Saptharishi, M.; Trebi Ollennu, A.; Dolan, J.; and Khosla, P. 1999. Cyber-ATVs: Dynamic and Distributed Reconnaissance and Surveillance Using All Terrain UGVs. In *Proceedings of the International Conference on Field and Service Robotics*, 329-334.

Turk, M.A., and Marra, M. 1986, Color Road Segmentation and Video Obstacle Detection, *In SPIE Proceedings of Mobile Robots*, Vol. 727, Cambridge, MA, 136-142.

Ulrich, I., and Nourbakhsh, I. 2000, Appearance-Based Place Recognition for Topological Localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, in press.