REAL-TIME OUTDOOR TRAIL DETECTION ON A MOBILE ROBOT

Andreas Bartel, Frank Meyer, Christopher Sinke, Thomas Wiemann,
Andreas Nüchter, Kai Lingemann, Joachim Hertzberg
University of Osnabrück, Institute of Computer Science
Knowledge Based Systems Research Group
Albrechtstr. 28, D-49069 Osnabrück, Germany
email: {nuechter|lingemann|hertzberg}@informatik.uni-osnabrueck.de

ABSTRACT
In this paper we present a reliable approach for real-time outdoor trail following and obstacle avoidance. The trail classification is done using an off-the-shelf webcam and a pitched 2D laser scanner on a KURT2 robot equipped with an Intel Centrino laptop. This simple setup enables us to follow given pathways of different kinds using a GPS receiver for rough orientation.

KEY WORDS
road detection and classification, road following, real-time mobile robot control

1 Introduction

A major challenge in outdoor mobile robotics is the classification of navigable and non-navigable terrain. The variability of the environment, which is caused by different types of trail surfaces, fuzzy road boundaries, and many distinct illumination conditions, pose great difficulties for reliable sensor data interpretation. In addition, autonomous robot navigation at higher velocities requires data interpretation to happen close to or at real-time. Driving at walking pace, for instance, while simultaneously avoiding hazardous situations such as leaving the trail, colliding with obstacles, approaching steep inclines or cliffs, demands fast controll cycles from the software. For autonomous robot navigation, two decisive issues have to be solved in parallel. Sensor data interpretation has to deliver meaningful results and this is expected to be done in real-time.

There are many ways how to approach this problem. During the DARPA (Defense Advanced Research Projects Agency) Grand Challenge in 2005, current state of art research on autonomous ground vehicle navigation, on a large scale, was demonstrated by many teams [5]. These competitors relied on high-end computer hardware and various different combinations of mulitple sensors such as 2D and 3D laser scanners, radar systems, and all kinds of vision systems. However, the processing capabilities of most autonomous mobile robots are usually limited to those of standard mobile computers. Therefore, it is of interest to investigate the possibilities and limitations of autonomous robot navigation that relies only on standard hardware and sensors. In that context, an important objective is to develop efficient algorithms that facilitate real-time processing of sensory information such that autonomous mobile robots are able to move at higher velocities and avoid any hazardous situations.

This paper presents a simple but efficient solution for autonomous outdoor navigation using a pitched 2D laser scanner and a webcam. Trail following at walking pace and avoiding of hazardous situations in real-time is achieved through statistical analysis of single 2D laser scanns and image processing. Laser data is being processed at full scan rate of 72 Hz, images are processed at 13 Hz. The modular software architecture is implemented on a KURT2 platform (see section 1.1). The system's performance is tested on trails within the Botanical Garden of the University of Osnabrück. The robot is able to navigate from a starting point to a goal position past defined GPS coordinates while remaining on the trail.

1.1 The KURT2 Robot

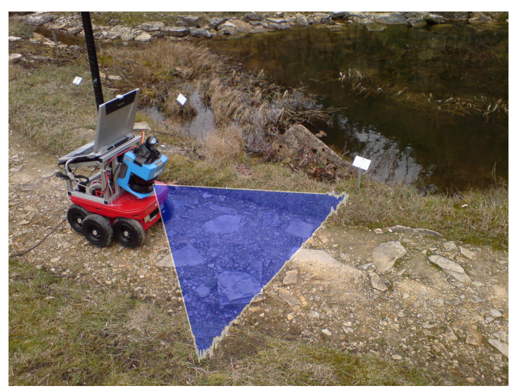


Figure 1. The KURT2 Robot with pitched laser scanner, webcam, GPS sensor and laptop.

KURT2 is a mobile robot platform with a size of 45 cm (length) $\times$ 33 cm (width) $\times$ 26 cm (height) and a weight of 15.6 kg. See Figure 1. Indoor as well as outdoor models exist. Equipped with a 2D laser range finder, the height increases to 47 cm and the weight increases to 22.6 kg. Two 90 W motors (short-term 200 W) are used to power the 6 wheels. The outdoor version has larger wheels compared to the indoor version, with the middle ones shifted outwards.

Front and rear wheels have no tread pattern to enhance rotating. Kurt2 operates for about 4 hours with one battery charge (28 NiMH cells, capacity: 4500 mAh). The core of the robot is an Intel-Centrino-1400 MHz with 768 MB RAM and a Linux operating system. An embedded 16-Bit CMOS microcontroller is used to process commands to the motor. A CAN interface connects the laptop with the microcontroller. The robot is equipped with three different sensors: A SICK laser range finder, a Logitech webcam and a Garmin GPS receiver. The laser scanner is fixed on the middle of the robot, tilted downwards in an angle of 23 degrees. A plane with 181 data points is scanned every 13 ms. The webcam is mounted on top of the range finder and delivers images with a resolution of 320 by 240 pixels at a framerate of 13fps. The 12 channel EGNOS enabled GPS receiver localizes the robot every second within the typical GPS error of about 10 to 15 meters.

1.2 Related Work

As shown in [4, 6], 2D laser range data on plain surfaces like gravel paths and cobblestone pavements varies significantly from data on grass, bushes and other plants, where range values are spatialy scattered. These principles are extendable to 3D laser point clouds [7–9]. However, 3D data analysis is computationally expensive and requires a 3D scanner or several 2D scanners.

A comprehensive approach to autonomous robot navigation using 2D laser data and visual information was recently introduced in [1–3]. Andersen et al. used a pitched 2D laser scanner for terrain classification. An algorithm was developed that fuses seven distinct classifiers that are obtained from a single scan: raw height, roughness, step size, curvature, slope, width, and invalid data. Long range support for road outline was provided via the analysis of chromacity and edge detection within single camera frames. The system was tested on a number of different roads, missions up to 3km in length have been completed successfully. Although many aspects of the approach reported here are similar to the work published in [2], there are two main differences. First, we prove that simple but efficient interpretation of distance-based mean and variance values is sufficient for navigable terrain classification of our test environment. Second, all of our sensors are being analyzed in real-time. Thus, we are able to interpret laser data at 72 Hz and video data at 13 Hz, which is quite an advancement compared to 10 Hz and 1 Hz as reported in [2].

2 Sensor Data Evaluation

Since the sensors work at different frequencies, a parallel software architecture is used in which each sensor is handled in a seperate thread. This avoids the polling problem in sequencial architectures and makes the system scaleable. After a sensor delivers new information, a suitable online classification algorithm is applied. Visual classification is based on standard image processing algorithms. The interpretation of the laser range data relies on statistical analysis. Efficient implementations for both purposes are provided by the open source library LTI-Lib.

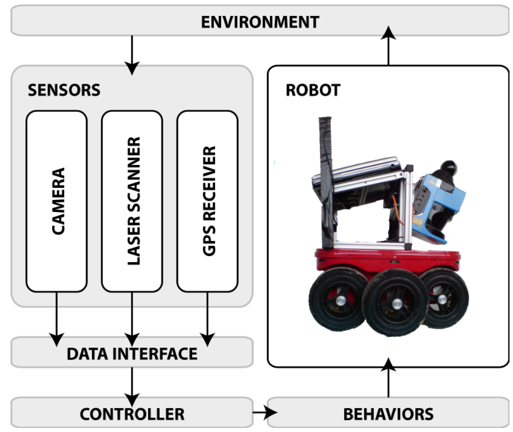


Figure 2. The software architecture of the KURT2 robot

2.1 Visual Classification

The algorithm consists of three main processing steps: Trail border detection, object extraction and direction control. Each processing step is divided into several sub-steps which are described in the following sections.

Between the steps of image processing, the CPU time slice is given to other sensor data processing threads and motor control. This ensures that after each step enough idle time is available for laser data classification and GPS localisation.

Trail Border Detection

Most trails have grass or planted borders. Therefore in a first processing step all green pixels of the image are replaced by black ones. This ensures that the contrast between the trail, which normally has gray or brown colors, and planted boundaries is maximized. Since LTI delivers color information in RGB format, we tried to set different RGB combinations to black. The best result is achieved if the following condition is fulfilled:

$$G \geq R \wedge G \geq B + 10.$$

Since color information is no longer needed after replacing green pixels, the following computations are performed on the gray scale channel.

In the next processing step, a Gaussian convolution is applied to eliminate fine textures and small structures like boundaries between cobblestones or gravel. Afterwards, the contrast is enhanced to emphasize the boundaries of the pathway and remove more irrelevant details. As a result of these processing steps, the pathway is repesented by the brightest area in the image. To seperate this area from the

Figure 3. Processing steps for border extraction: The figure shows an original webcam image and the steps after applying the following filters: green to black, gaussian blur, contrast enhancement, thresholding and border extraction.

rest of the image, a threshold filter is applied that paints black all areas that do not belong to a pathway.
Finally, a gradient filter is applied to extract the edges of the detected pathway. The results of these processing steps are shown in Fig. 3.

Object Extraction

Sometimes the boundaries of disruptions like shadows or highlights are marked by the edge extraction algorithm. In most cases, these wrongly classified structures lie within the boundaries of the trail or interrupt them. Since the camera is slightly shiftet downwards, the path fills most of the camera image. Therefore the biggest contour surrounding the smaller ones is extracted using the LTI-Libs object extraction algorithm (cf. Fig. 4).

Direction Control

To generate a control signal, the robot's position on a trail is estimated as follows: A vertical line in the center of the picture is used as a reference. The intersections between a horizontal line and the extracted boundaries are calculated. The centre of the intersections is compared to a point on the reference line. On the center of the trail, the two points should match. Otherwise, the calculated point is shifted with respect to the reference point. This shift is interpreted as a control signal: By driving in the direction of the computed shift, the robot will move towards the center of the trail.

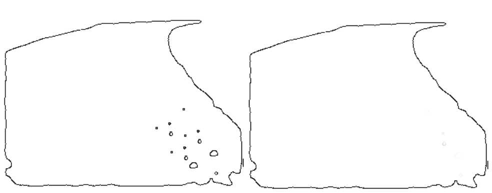


Figure 4. Detected borders after edge detection (left) and object extraction(right).

In order to follow the course of the trail, we compute the deflection on several horizontal lines. The final control command is computed by calculating the weighted mean of these deflections. Since lower lines represent the trail directly in front of the robot, these lines have a greater weight than the upper lines (see Fig. 5).

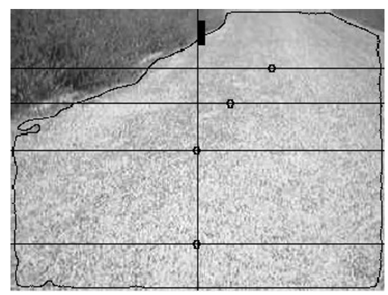


Figure 5. Signal Generation: Horizontal lines corresponding to four increasing qualitative distances to the robot are used for generating the direction control signal.

2.2 Laser Scanner Data Classification

The laser data classification algorithm is based on statistical analysis of the provided proximity values. Since only the area in front of the robot is of interest, the field of view of the laser scanner is restricted to 40 degrees on the left and right side. The mean and the variance of the proximity values within each area are calculated. The classification algorithm uses these four values to decide in which direction to drive. Since these operations only need little computational time, the robot is able to analyze the scans at a frequency of 72 Hz. This high classification rate allows the robot to quickly detect hazards such as potholes, downward stairs and obstacles in front of it.

Avoiding Hazardous Situations

Hazard situations are detected using the mean of the proximity values. If the mean exceeds a predefined maximum, the robot assumes to be in front of an unpassable hole in the

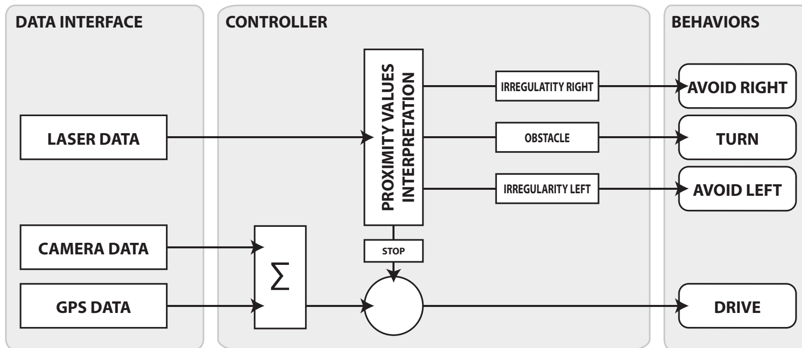


Figure 6. The robot control architecture: If the laser data interpretation reveals a hazardous situation in front of the robot, the driving behaviour is stopped and a suitable avoiding behavior activated. Otherwise the camera signal is merged with the GPS signal to follow the given trail.

ground. If the mean falls below a certain minimum value, an obstacle in front of the robot is detected. In both cases a suitable turn behaviour is activated.

Classification of Navigable Terrain

The trail classification relies on the assumption that navigable terrain are relatively plain. Therefore it is expected that the variance of the proximity values in these regions is lower compared to non-navigable areas. In non-navigable regions, objects on the surface, plants or potholes will increase the variance of the proximity values.
In order to generate a control signal, the laser scanner values are divided into two parts representing the left and right side of the trail profile in front of the robot. If the variance on one side exceeds a threshold and the variance on the other side is below this threshold, a control signal towards the area with the lower variance is generated. As a result the robot will navigate towards regions of low variance. Sample scans for hazardous situations and trail detection are shown in Fig. 7

2.3 GPS Guidance

While the robot is moving, its current position is determined using the onboard GPS Sensor. The difference between two consecutive positions is used to calculate the current bearing towards the next given GPS waypoint. The left and right components of the bearing indicate in which direction the robot has to turn.

3 Robot Control

A controller module merges the different control signals derived by the three sensors into one final controlling output. To achieve an adequate hierachy of importance, four basic behaviours are implemented: A driving behavior, two behaviors for avoiding non-navigable areas on the left or right side of the robot, and a turn behaviour to drive away from obstacles.

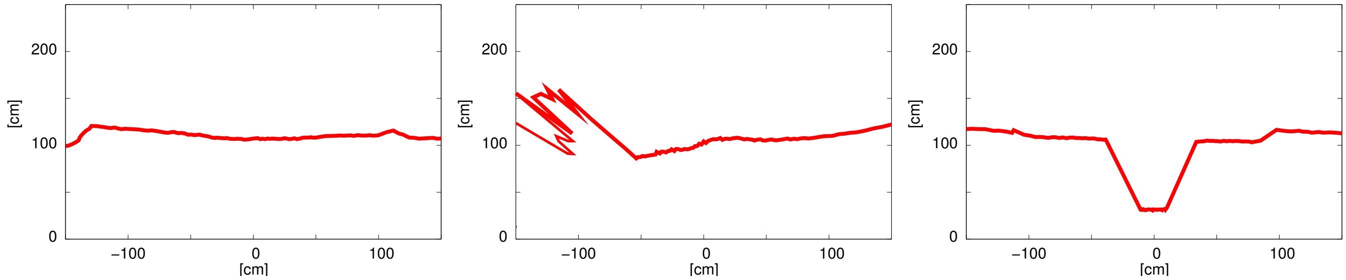
Since detecting hazardous situations is of great importance, the derived signal by the distance-based obstacle detection gets the highest priority. If the center of the moving direction is classified non-navigable, the robot makes a full stop and switches to the turn behaviour. The first action of this behaviour is to set back one robot length. Then, if the area in the direction of the next waypoint is classified as driveable, the robot turns into this direction and resumes normal driving behaviour. Otherwise, it turns towards the direction with minimal variance in the laser range data.

If the surface is classified non-navigable either on the left or right side of the moving direction, the robot switches to one of the avoiding behaviours. The resulting control output is a short move into the opposite direction.

The driving behaviour will be active if no hazards are detected by the laser scanner. In this case the control signals from the visual path classification and the navigation module are fused into one output signal. The resulting signal keeps the robot on the trail that leads to the next goal. The influence of the GPS signal depends on the robot's distance to the next way point. As long as the next waypoint is still far away, the output signal is generated by the visual classification steps. If the robot gets closer to the next waypoint, the GPS signal will increase its influence and turn the robot towards the right direction. The complete control architecture is shown in Fig. 6.

4 Experiments

The performance of the system was tested in the Botanical Garden of the University of Osnabrück. There are a wide variety of trail types; some are plain and have clear defined boundaries, others are rough and consist mainly of cobblestones or are covered with gravel. Please refer to `http://kos.informatik.uni-osnabrueck.de/download/kgc/` for videos showing the autonomous behaviour of the KURT2 robot.



(a)

(b)

(c)

Figure 7. Some sample laser scans taken on the same pathway in the botanical garden. Scan (a) was taken in the plain middle of the road. Scan (b) was taken on the left border and shows a high variance in the left part of the range data. Since the variance in the right part is low, a control signal in that direction was generated. Scan (c) shows an obstacle directly in front of the robot. After that scan the turn-behavior was activated.

Figure 8. Different kinds of trails used for reliability evaluation: A graveled trail bordered by a stone wall (top left), a trail surrounded by lawns (top right), a cobblestone trail (bottom left), and the main trail of the Botanical Garden that is bordered by different types of terrains (bottom right).

Table 1. Correct boundary detections for each sensor and the fused control signal on different trails.

Trail	Data Sets	Scanner	Camera	Signal
Wall	5590	98 %	0 %	98 %
Lawn	6030	27 %	92 %	98 %
Cobblestone	4410	59 %	30 %	85 %
Main Trail 1	7452	54 %	61 %	91 %
Main Trail 2	17482	46 %	55 %	91 %

Reliability Evaluation

To evaluate the reliablity of the different classification algorithms and the fused data compared to "ground truth" we set up the following experiment: The robot was driven manually on one boundary of a trail. In this position, each sensor is expected to generate a control signal towards the middle of the trail. These control signals were logged and compared with predefined values. To cover the different kinds of terrains, we have chosen four characteristical trails that are predominant in the botanical garden (see Fig. 8). The results of these experiments are shown in Table 1.

The first tested trail was covered with light gray gravel bordered by a quarrystone wall. Since the contrast between the trail and the wall was very low, the camera was unable to detect the trail's boundary. However, the vertical alignment of the wall resulted in descreasing distance values and increasing variance within each laser scan. Therefore it was possible to recognise this wall as an obstacle and detect the trail almost perfectly.

The second trail was plain and bordered by lawn. Here the grass was nearly cut down to the level of the trail. Therefore, in most cases, the border between the lawn and the trail was not recognisable within the laser data. Since the green replacement step in the visual classification algorithm results in a contrast difference between the trail and the green areas, the camera signal was accurate in 92%. The fused signal reached an accuracy of 98%.

The third trail was a small cobblestone path. In this case, the boundary of the path was not clearly defined. Since we tried to follow the obvious border of the paved areas, our given ground truth did not represent the boundary of non-navigable regions. Hence, the generated control signal did not always match our expectations, resulting in a rather low fused accuracy of 85%. Even though the robot left the cobbled road during autonomous driving mode, it never collided with any obstacles on this trail or drove into obviously non-navigable areas. This case demonstrates that the overall system performance is far better than indicated by this experiment.

The last two tests were performed on longer sections of the main trial in the Botanical Garden. We chose sections with different kinds of boundaries. Since each sensor has classification disadvantages in some border configurations, the overall stand-alone performance of every single sensor was mediocre. The fusion of the camera and scanner signals lead to accuracies of 91%.

The results of these experiments show that visual classification detects nearly all bounderies of the garden's trails correctly in situations where contrast between the pathways and their borders is high. Severe problems occur if the field of view of the camera is outside a trail. In these cases, other objects, which stand out in the background, are classified as driveable areas. Also stony walls tend to have high contrast differences between the stones, which are certainly not

the road. In such cases, the algorithm classifies the splices as boundary, corrupting the signal. Under bright sunlight, hard shadows and highlights generate high contrast differences in the camera images, falsifying the detected road borders. In these situations, the robot tends to avoid the shadows.
The laser scanner classification is most reliable. Collisions even with moving objects like walking persons never happened. Since in most cases the changeover from driveable to non-driveable is characterized by ridges, the statistical interpretation of range data works almost perfectly. Problems occured if the pathway was bordered by lawn areas where the grass was on the same level as the road.
Although both approaches sometimes mis-classify the trails, the interaction of both leads to an appropriate performance. If the robot drives towards ridges or other obstacles, the control signal from the laser scanner classification initiates a turn. Low boundaries on the other hand are often detected by the camera owing to the green pixel replacement.
The results of the tested trails show high reliablity rates of 80 to 90%. But these values do not reflect the practical performance of the system. In experiments, the robot was manually forced to drive on a border of a trail. Since in autonomous driving mode, it would avoid those borders and stay on the trail, it would have had several chances to detect a boundary before it reaches the actual border. Therefore the error rate in autonomous mode is far lower than the experimental results suggest.
The performance of the robot in autonomous driving mode is demonstrated in the provided videos. The videos show that the robot was able to follow any given trail at walking pace and avoid possible hazardous situations. In more than 30 minutes of autonomous driving, only two manual direction corrections via joystick were needed. However, in some cases, when a trail got too bumpy, we had to stabilize the robot in order to avoid a tip over, since the robot is not well-balanced with the heavy laser scanner mounted on the front side.
With activated GPS navigation, the visual classification was ruined. In small scale environments like the Botanical Garden, the GPS resolution of 10 to 15 meters is too inaccurat for reliable navigation. In most cases, the target way points were detected outside the trail, therefore the robot tended to head towards non-navigable areas. Only on very broad pathways was the robot able to navigate into the correct direction.

5 Summary and Outlook

This paper has presented an approach for outdoor road following. The resulting system is highly reliable in keeping the robot on the road as well as avoiding dangers, i.e. obstacles, potholes and downward steps. Nevertheless the hardware requirements for the underlying computations are very low so that the system is open for functional expansions.
The sensors used for road classification were a standard webcam and a 2D laser scanner. All computations were done online on an Intel Centrino 1400 MHz laptop.
The system was tested in the Botanical Garden of the University of Osnabrück. In autonomous driving mode the robot was able to follow different kinds of roads including graveled and roughly cobbled pathways. The evaluated performance showed reliability rates beyond 90%. The experienced overall performance was even much better.
Given its parallel software architecture, the system is suitable for the newly developed multi-core processors like the Intel Core Duo. On these multi threading systems, more computational time will be available providing the opportunity to implement more sophisticated algorithms using learning methods for visual classification or parameter tuning.

Acknowledgements

This research was partly funded by the European Commission's 6th Framework Programme IST Project MACS under contract/grant number FP6-004381. The Commission's support is gratefully acknowledged.

References

[1] J. C. Andersen, N. Andersen, and O. Ravn. Vision assisted laser scanner navigation for autonomous robots. In *10th Int. Symp. on Experimental Robotics 2006 (ISER '06)*, 2006.

[2] J. C. Andersen, M. R. Blas, N. Andersen, O. Ravn, and M. Blanke. Traversable terrain classification for outdoor autonomous robots using single 2D laser scans. *Integrated Computer-Aided Engineering*, 13(3):223–232, 2006.

[3] M. R. Blas, S. Riisgaard, O. Ravn, N. Andersen, M. Blanke, and J. C. Andersen. Terrain classification for outdoor autonomous robots using 2D laser scans. In *2nd Int. Conf. on Informatics in Control, Automation and Robotics, ICINCO 2005.*, 2005.

[4] M. Castelnovi, R. Arkin, and T. R. Collins. Reactive speed control system based on terrain roughness detection. In *Proc. Int. Conf. on Robotics and Automation (ICRA '05)*, 2005.

[5] S. Thrun et al. Winning the darpa grand challenge. *Journal of Field Robotics*, 23(9):661 – 692, August 2006.

[6] J. Macedo, R. Manduchi, and L. Matthies. Ladar-based discrimination of grass from obstacles for autonomous navigation. In *ISER '00: Experimental Robotics VII*, 2001.

[7] A. Nüchter, K. Lingemann, and J. Hertzberg. Extracting drivable surfaces in outdoor 6d slam. In *Proc. of the 37nd Int. Symp. on Robotics (ISR '06)*, Munich, Germany, 2006.

[8] S. Thrun, M. Montemerlo, and A. Aron. Probabilistic terrain analysis for high-speed desert driving. In *Proc. of the Robotics Science and Systems Conference*, 2006.

[9] Nicolas Vandapel, Daniel Huber, Anuj Kapuria, and Martial Hebert. Natural terrain classification using 3-d ladar data. In *IEEE International Conference on Robotics and Automation*, volume 5, pages 5117 – 5122, April 2004.