Reactive, Safe Navigation for Lunar and Planetary Robots

Hans Utz*
NASA Ames Research, Moffett Field, CA, 94040, USA

Thomas Ruland†
University of Ulm, Ulm, 89073, Germany

When humans return to the moon, Astronauts will be accompanied by robotic helpers. Enabling robots to safely operate near astronauts on the lunar surface has the potential to significantly improve the efficiency of crew surface operations. Safely operating robots in close proximity to astronauts on the lunar surface requires reactive obstacle avoidance capabilities not available on existing planetary robots. In this paper we present work on safe, reactive navigation using a stereo based high-speed terrain analysis and obstacle avoidance system. Advances in the design of the algorithms allow it to run terrain analysis and obstacle avoidance algorithms at full frame rate (30Hz) on off the shelf hardware. The results of this analysis are fed into a fast, reactive path selection module, enforcing the safety of the chosen actions. The key components of the system are discussed and test results are presented.

I. INTRODUCTION

Improving navigation capabilities is central for advancing robotic applications for future planetary and lunar missions. Advanced, safe robot navigation systems open up mission operations scenarios that increase efficiency and lower mission risk. When humans return to the moon, they will be accompanied by robotic aids, helping them to cope with this harsh environment. Current research at NASA for lunar robot applications encompasses a variety of robotic platforms, ranging from a robotic habitat, over a robotic successor of the lunar roving vehicle (LRV) to small payload robots that can carry scientific instruments or perform simple maintenance tasks. For lunar missions, enabling robots to safely operate near astronauts on the lunar surface has the potential to significantly improve the efficiency of crew surface operations. Such an operational scenario poses requirements to the navigation systems of these robotic vehicles, not met by today's planetary rovers.

Operating robots in the proximity of astronauts poses a new challenge for space robots, as this breaks the assumption of a static world, where the robot is the sole actor and the position of obstacles can not change over time. It also requires far higher operational speed from the robots to keep up with the speed of the astronauts. Furthermore, the definition of safe robot motion is no longer restricted to "safe for the robot", but includes safety of astronauts or other surface assets. Safely operating robots in an environment with moving (dynamic) obstacles imposes two crucial changes with respect to current space-robotic navigation capabilities. (1) Robots will have to update their world model and react on this new information at frequencies typically above the capabilities of most deliberative planner-based systems. (2) Nevertheless, the adherence to safety constraints needs to be easily verifyable, a feature that reactive systems often are blamed for missing.

In this paper we present a work on safe, reactive obstacle avoidance in unstructured, unvegetated terrain, using a stereo based high-speed terrain analysis and obstacle detection system on the K10 robot platfrom.[1] Stereo correlation is performed in hardware, using the Tyzx G2 sensor,[2] which can provide depth images at 30Hz. The system is capable of analyzing the terrain for traversability (and obstacles) at full frame rate using standard COTS PCs. This "High Frequency Terrain Analysis" (HFTA) information is evaluated in a "safe tele-operation" scenario. It is based on a reactive control scheme, that features a fast control loop, while allowing to ensure safety constraints.

The remainder of this paper is organized as follows. In the following section an overview of the target robotic system as well as the target scenario is given. In section III related work on terrain traversability is

*Staff Scientist, RIACS/USRA, Moffett Field, M/S 269-3.
†Student, Neuroinformatics, University of Ulm.

discussed, before our approach is presented in section IV. Afterwards related work on safe navigation as well as our realization is discussed, before experimental results are presented in section VII. The paper closes with conclusions and the identification of future work.

II. SYSTEM OVERVIEW

In this section the hardware platform and the target system capabilities are discussed. The Intelligent Robotics Group (IRG) at NASA Ames is dedicated to improved understanding of extreme environments, remote locations, and uncharted worlds. IRG conducts applied research in a wide range of areas with an emphasis on robotics system science and field testing. Current applications include planetary exploration, human-robot fieldwork, and remote science.

II.A. The K10 Rover

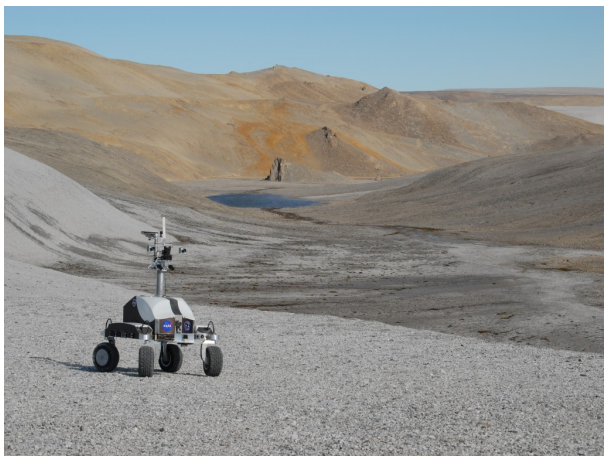


Figure 1: K10 on "Drill Hill" at Haughton Crater on Devon Island, Canada (©Matthew Deans, Haughton Mars Project 2007).

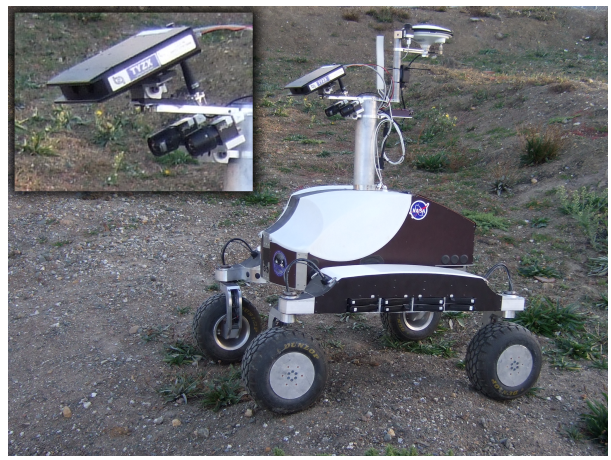


Figure 2: K10 carrying the Tyzx G2 3D sensor on Marscape.

The primary research platform of the IRG is the K10 rover platform. K10 is a 4-wheeled small payload rover, with roughly 1m × 1m footprint. It is equipped with a 4-wheel-drive/4-wheel-steering locomotion system for continuous driving with a maximum velocity of $0.85\frac{\mathrm{m}}{\mathrm{s}}$ (about human walking speed). The rover is manufactured mostly from commercial off-the-shelf parts and its on-board computational resources consist of a Dual-Core 2GHz Laptop with 2GB of RAM. Fig. 1 illustrates the K10 rover at a field test at Haughton Crater in the Canadian High Arctic.

II.B. Tyzx DeepSea G2 3D Sensor

For obstacle detection in rough unstructured terrain, precise information about the environment is required. In this work, a commercial *Tyzx DeepSea G2 Vision System* is used to acquire 3D information. This system provides a complete solution for binocular stereo vision in a single, small size device. It includes two grayscale cameras to acquire images of the scene in front of the robot from two slightly different points of view. An integrated circuit, designed particularly for stereo image processing, recovers depth information based on the arising differences in the recorded images. Depth images from stereo-vision were previously computed in software on the K10 rover. It has already shown to work especially well in the highly textured environment of Marsscape. In numerous experiments, it demonstrated to be a completely appropriate solution for day-light environment perception. Because of the significant computational complexity of the depth reconstruction process, only a limited frame-rate ($\approx$ 2Hz) could be achieved on general purpose computing hardware.

This dedicated hardware implementation allows the retrieval of 500×312 3D measurements at a frame-rate of 30Hz. The G2 has a base-line of 6.00cm and features sub-pixel correlation. This results in a very good accuracy of the depth image information, with respect to the target application. In this work, a maximum look-ahead distance for obstacle avoidance of 5m was used, at which the sensor shows a standard deviation

of 45mm. The horizontal field of view of 62.6° is quite narrow for the desired application, but did not restrict the algorithmic solutions in the application scenario.

Fig. 2 shows K10 Black on "Marscape", a Mars-analog rover test facility at NASA Ames Research Center. The cutout shows the Tyzx G2 sensor (black box) mounted on K10's mast. The cameras mounted below the G2 are part of the existing software stereo vision based obstacle detection system.

II.C. Target Scenario

The target scenario is a robot working in close proximity of humans, as demonstrated for instance in.[3] For the test application a safe-guarded tele-operation system was chosen. This setup allows interactive testing of the robot's avoidance capabilities, allowing to process a large variance of test cases with little additional effort. The modularity of the control loop was an important design goal though, as the the obstacle avoidance capabilities are to be flexibly combined with other configurations, such as way-point or task-oriented navigation. The targeted run-time of the safe navigation system was to fit the entire control loop into the 33ms time-slot of one depth image frame, thus allowing the optimal reactivity to the sensory source.

III. Terrain Traversability Analysis

In recent years, navigation in rough terrain gained increasing attention. Planetary rover projects, like *Path Finder* of NASA JPL in 1997[4] or the still successful twin rover operation of the 2004 *Mars Exploration Rover*[5,6] mission gathered broad publicity. Another driving force in 2004 was the announcement of the DARPA Grand Challenge. Mobile vehicles (mostly cars) were supposed to traverse about 240km of cluttered dust-road path.

Terrain traversability analysis (TTA) in dust-road scenarios differs significantly from obstacle avoidance in-doors or urban environments, which were studied in the beginnings. The lack of highly structured components complicates the problem. This is especially highlighted by the 2004 Grand Challenge result, with all of the competitors failing in the first 5% of the overall travel-distance. The specific geometry of the terrain now has to be analyzed. Simple thresholding of input values is not sufficient anymore.

Remote, hazardous environments like planetary surfaces pose an additional challenge, as no roads exist, that can aid in navigation decisions. Navigation under such conditions, requires knowledge about areas of safely traversable terrain, to base path selection on. This information is typically organized in *traversability-maps*, which hold locations of terrain, which potentially are hazardous to traverse for the robot. Usually, traversability-maps may distinguish between terrain patches, which should be avoided at any costs to prevent damage, and other areas, which could be passed safely by reducing translational velocity. This enables navigation to make more efficient decisions. Trade-offs between long detours and traversal of slightly cluttered areas at reduced speed are made possible (see[7]).

This work is focused on operation in *dynamic environments*, which introduces further requirements. For safe navigation, fast reactions are crucial. Obstacle maps have to be kept up-to-date at high frequency. They do not benefit very much from a-priori information like satellite imagery, which had been gathered at a previous point in time. Thus, maps have to be computed from the available sensor data in real-time.

At the same time, traversability information of the terrain, which the vehicle is about to traverse, needs to be available at a *high resolution*. A lack of resolution forces conservative estimates and decisions that can result in unnecessary detours of the rover. A minimum of resolution also has to be provided to detect even small obstacles and pits, which still might endanger the rover. This high resolution is reported by[8] to be unlikely available in a-priori elevation maps, hence hazardous objects have to be inferred online from on-board sensor data.

III.A. An Ideal Algorithm

As mentioned above, terrain analysis tries to predict dangers, which arise from specific terrain features. An ideal algorithm could predict the vehicle's behavior on the perceived terrain, based on an actual physics simulation An ideal physics model could consider all effects and interactions of the rover with the underlying terrain. For example tire slippage on sandy ground, tip-over as a result of driving a too steep slope, or holes which could grab hold of a wheel. This way, every potential terrain hazard could be anticipated and thus avoided by the navigation software.

Unfortunately, on today's computing hardware, such a simulation can not be performed in real-time, considering the required precision. So instead statistical analyses are used to detect specific terrain features, which are known to lead to unsafe configurations. Domain knowledge is applied to design more efficient algorithms, which fulfill precision and speed requirements.

III.B. Baseline Algorithm MORPHIN

The baseline of this work was the MORPHIN[9] algorithm, as currently used on the K10 rovers. MORPHIN processes geometrical terrain information, gathered by any 3D sensor. The measurements are collected in a *Digital Elevation Map* (DEM). This map is segmented into overlapping, square *patches* of 2.1m side length. The patches feature a relative displacement of 0.3m in both ground dimensions. This configuration creates a grid layout of *cells* with 0.3m side length (illustrated in Fig. 3). Due to the overlapping alignment, each cell contributes to the area of 49 patches. The complete grid layout is defined statically against the world coordinate frame.

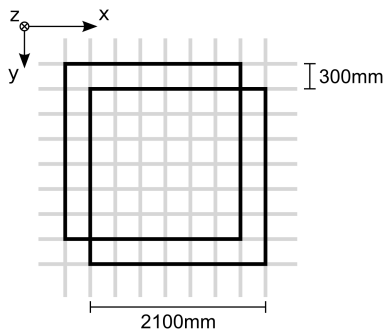


Figure 3: Overlapping rover sized terrain patches in the digital elevation map.

For efficient processing, each 3D measurement is assigned to one of these cells according to its position $(x, y)^\mathsf{T}$ in the world frame. MORPHIN thus performs a back-projection and transformation of the complete input data, from the sensor to the world coordinate system. For each *patch* the traversability is assessed, using two measures, that are based on the statistical properties of all measurements from the depth image, whose $(x, y)^\mathsf{T}$ world-coordinates lie with the patch: (1) The slope, which is the angle of the plane fitted through the input point-cloud. (2) The roughness, which determines the quality of the plane-fit, using the mean square error. The two measures are combined in a weighted sum and stored in the traversability map. Thresholding is used to mark unsafe terrain.

In numerous field-tests, MORPHIN evaluated for off-road terrain traversability analysis. It had been ported to a broad range of mobile vehicles (K9, K10, ATRV2, Hyperion) and thereby contributed to the success of various projects.[7,10,11] Although delivering qualitatively remarkable and reliable obstacle detections, the profiling results on the K10 hardware showed up one major disadvantage – 105ms of computation time. By exceeding the envisioned maximum cycle time of 33ms (update frequency of 30Hz), an application in the targeted highly dynamic scenario using a behavior based control-loop is not directly possible.

IV. HIGH FREQUENCY TERRAIN ANALYSIS

In this section we present a short overview on the main algorithmic advances of HFTA over the baseline algorithm. Given the envisioned run-time of the HFTA, it was obvious, that such a reduction in computational demands was not possible by merely engineering a faster implementation.

IV.A. Performance Analysis

A theoretical analysis of MORPHIN revealed several hot-spots in the algorithm, that were responsible for the majority of run-time of the algorithm.

1. The initial coordinate transform, even though implemented very efficiently, is extremely computationally expensive.

2. The whole input data set (the point-cloud), is carried through almost all of the analysis process.
3. The implementation of the cell-values per patch summation features a quadratic complexity $O(n^2)$ with respect to the patch-side length.

IV.B. HFTA Improvements

In order to reduce the run-time of the terrain traversability analysis for use at the full frame rate of the depth image stream, all of the above bottlenecks were addressed.

Cell assignment: HFTA avoids the costly initial coordinate transform to the world-coordinate frame, by doing the cell-assignment and patch analysis in the image coordinate system. Only the slope measure for each patch needs to be transformed into the world-coordinate frame. The fixed mounting of the sensor, as well as it's axis-parallel view result in a less complex, fixed transformation, which can be precomputed and efficiently implemented with the use of look-up tables.

Early data set reduction: Instead of using the whole input point cloud for slope and roughness, HFTA calculates a single representative for each cell and calculates the plane-fit as well as the plane-fit-error for each patch, only based on the representatives of the contributing cells. To compensate for the reduced sample number, the cell-size is reduced to 0.1m.

Cell fusion: The highly efficient plane fitting algorithm of MORPHIN calculates various parameters per cell, that are combined for the plane-fit of a terrain patch. Combining the individual measurements of the cell is implemented using a simple summation over the patch square. This features a computational complexity of $O(n^2)$ with respect to the square width. As HFTA works on a smaller cell size and therefore combines more cells per patch, this bottleneck was replaced by an efficient box-filter based implementation, which computes the summation with linear complexity ($O(n)$).

IV.C. Performance comparison

The algorithmic improvements of HFTA led to a computation time reduction of 88%, when compared to the baseline system MORPHIN. The processing time per depth image, have been confined to 12ms on the same hardware platform. The results of the traversability evaluation though are of comparable quality, as verified in a set of qualitative evaluations.

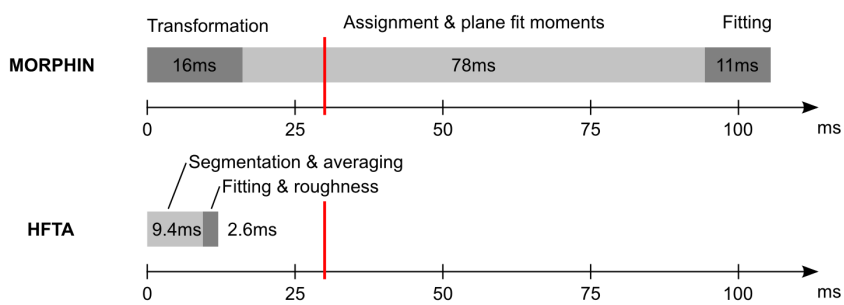


Figure 4: Computation time comparison between MORPHIN (top) and HFTA (bottom). The upper 33ms boundary, which has to complied with in order to enable the envisioned update frequency of 30Hz, is illustrated by a red line.

V. SAFE NAVIGATION

In order to navigate safely the obstacle information stored in the goodness map needs to be translated into a series of locomotion commands for the robot. In a static environment a global motion planner can be deployed for obtaining an optimal solution. In the absences of complete a-priori information, such as the targeted dynamic uncontrolled environment, the motion plan would need to be updated frequently. The time-constraints of the target scenario prohibit the run-time associated with global optimization on

this detail level. In such scenarios, reactive, behavior-based approaches or one-shot planners are generally favored. They can not guarantee global optimality anymore. Instead a greedy algorithm with limited look-ahead is applied, that only ensures the adherence to local constraints, such as collision safety. In this work, the modularity of the reactive control loop was also of central concern. The goal was, to be able to combine the obstacle avoidance module with different control schemes, such as waypoint navigation or task-oriented motion, without having to re-verify the safety of the resulting combined control loop.

In the following section, *local* and *global* strategies for obstacle avoidance will be compared according to the target scenario of this work, before the design of the reactive control scheme is developed. Exemplary realizations will be the D^* algorithm, which is part of K10's current setup,[12] and the *GESTALT* local avoidance planner.[5]

V.A. K10's current Obstacle Avoidance

The current obstacle avoidance on-board the K10 rovers is conducted by the *global* navigation module. In this setup, no separate obstacle avoidance module exists. Instead, avoidance is performed implicitly by global navigation. In this process, a set of short, predetermined arcs is evaluated concerning local and global cost.[11] Local cost is given by the terrain traversability along each arc while global cost is a measure of path length between the end of the arc and the goal. This path is determined by the D* algorithm as described in.[12]

This configuration promises the most efficient, goal directed paths since obstacle avoidance thereby makes globally optimal decisions. On the other hand, global path planing is computationally complex and thus unlikely to be executable at the desired cycle frequency of 30Hz. An application in the targeted dynamic scenario is thus not directly possible.

V.B. GESTALT

In 2002, the NASA Jet Propulsion Laboratory presented a new *local* obstacle avoidance planner.[5] It was developed previous to the Mars Exploration Rover missions in order to be used on a variety of robotic platforms. GESTALT combines both major navigation tasks, terrain traversability analysis and obstacle avoidance and is targeted towards application in a *static environment*. It's TTA module, is based on the MORPHIN algorithm and also utilizes input from passive stereo vision. In the following, the latter obstacle avoidance part will be focused on.

GESTALT receives navigation goals as static way-points in the world coordinate frame. It selects movement commands based on these way-points and the current state of the world. The local traversability map is described to be a *configuration space*. The goodness value of each cell denotes, whether a rover-sized object centered at that cell would be endangered by an obstacle anywhere underneath its base area. This implicates, that the cell goodness is also influenced by obstacles, which are located outside the cell. Obstacles are thus represented by half the rover-size larger than their real world dimensions.

Based on this configuration space, safe movements get selected. The *action space* for that selection consists of a discrete subset of all circular arcs (0.35m in length), which can be executed by the robotic vehicle. Since no continuous motion is intended, the current state of the rover does not limit this action space. The finite number of executable arcs is evaluated concerning hazardousness and goal progress. The hazardousness of an arc is based on the cells, which are traversed on the way up to the look-ahead distance. In addition to the safety of the arc, the progress towards the goal is taken into account for scoring the utility of an arc. Both ratings are then fused for each arc. The arc with the maximum voting score is finally selected and executed. During the 0.35m traversal, no further observation is conducted.

V.C. Selected Approach

Due to the strict requirements concerning computation time and reaction delays, a local, behavior based obstacle avoidance approach had been chosen for this work. Since map maintenance over time is not directly possible in a dynamic scenario, locally optimal motion-decisions will be made exclusively on the basis of currently available sensor data and its analyzed traversability. The selected reactive control scheme generalizes the evaluation models that D* and GESTALT provide, taking a behavior oriented point of view. The process can be split into three fairly isolated stages, command generation, command evaluation and command selection.

Initially a set of possible actions (motion-commands) is generated for further evaluation by the system, based on the task at hand and the action constraints of the robot. Classical behavior based approaches like potential fields of fuzzy-sets try to optimize over the whole action space. The dynamic window approach reduced this working set to the action space actually accessible given the dynamic constraints of the robot and the time-horizon, given by the evaluation frequency of the control loop. Pre-selecting an arbitrary set of actions allows to further reduce the size of the input data, especially for robots with more degrees of freedom.

During command evaluation, a set of modules rates the commands along different metrics, such as goal-directedness or safety. Apart from rating, each module can also veto an action, excluding it from further consideration. In a final step the arbitration module combines the individual ratings and selects from the non-vetoed actions the one with the best score.

VI. EXPERIMENTAL SETUP

As a test application, a *safeguarded tele-operation* scenario was chosen, using a joystick as input device. The robot was controlled by an operator from the command trailer with a joystick driving up to $0.40\frac{\mathrm{m}}{\mathrm{s}}$, but autonomously avoiding any obstacles, when they reached into the commanded path. The Tyzx G2 3D sensor was mounted on-board the K10 rover and the HFTA module has been fully integrated with the *Service Oriented Robotics Architecture*,[13] which is operating the K10 rovers. The measurement stream from the sensor was delivered by the *Miro*[14] middleware infrastructure. The joystick scenario was chosen, as it most easily allows to stress test the system with lots of potentially hazardous maneuvers. Another application would be to combine the avoidance module with path planning modules, for autonomous navigation.

Command Generation: The generated commands depend on the input, which was requested by a higher-level control module (e.g. navigation, user interface). This request should be executed whenever possible. In case a hazardous obstacle prohibits the resulting path, alternative motions have to be available. These alternatives, should resemble the requested command as close as safely possible. To achieve this behavior, a preference indicator is assigned to each motion. The indicator is highest for the requested command and declines as alternatives deviate more from the input.

The command alternatives are generated with increasing absolute deviations from the requested rotation velocity up to a certain threshold. This is repeated several times with decreasing translation velocities until the robot is rotating in place.

Command Evaluation & Selection: For each suggested command, the resulting trajectory of the robot is predicted for evaluation. A simple arc-base motion model was assumed, ignoring the limited steering speed of the rover. This obviously is just a rough approximation of the actual physical rover movement, and higher safety distances had to be selected to accommodate for the lack of precision of the motion model.

Using this arc-model, the resulting path of all possible commands gets predicted. The length of each arc is determined by the rover's stopping distance at the corresponding translation velocity plus a safety margin of 1.00m. The path is then sampled onto the configuration space to evaluate its goodness on the basis of the traversed cells. Similar to traversability, the command/path goodness is given by an interval measurement ($[0, 1]$). Again a goodness of 1 denotes safely traversable terrain while 0 vetoes the execution of a command.

The goodness evaluation function is kept fairly simple to ensure short computation times. This enables the examination of a large number of potential paths and thereby enables precise navigation. Goodness is simply calculated by the minimum traversability of all configuration space cells, which are passed by the path.

If a path's goodness is less than a traversability threshold this implies that the path crosses an impassable cell. Since its length is given by the rover's stopping distance, having a severely dangerous obstacle on the predicted configuration space path would cause the rover to hit upon it with some part of its chassis. The executed command thus has to be selected among those which produce a path featuring a goodness greater than the selected threshold. The winning command is finally determined by the maximum preference indicator over this set of non-hazardous alternatives. This behavior ensures the execution of a command, which is as similar as possible to the requested input.

The actual goodness of the winner could be used to adjust the translation velocity, which is used to traverse the path. Thereby, rough areas could be approached more cautiously. This behavior is not yet implemented.

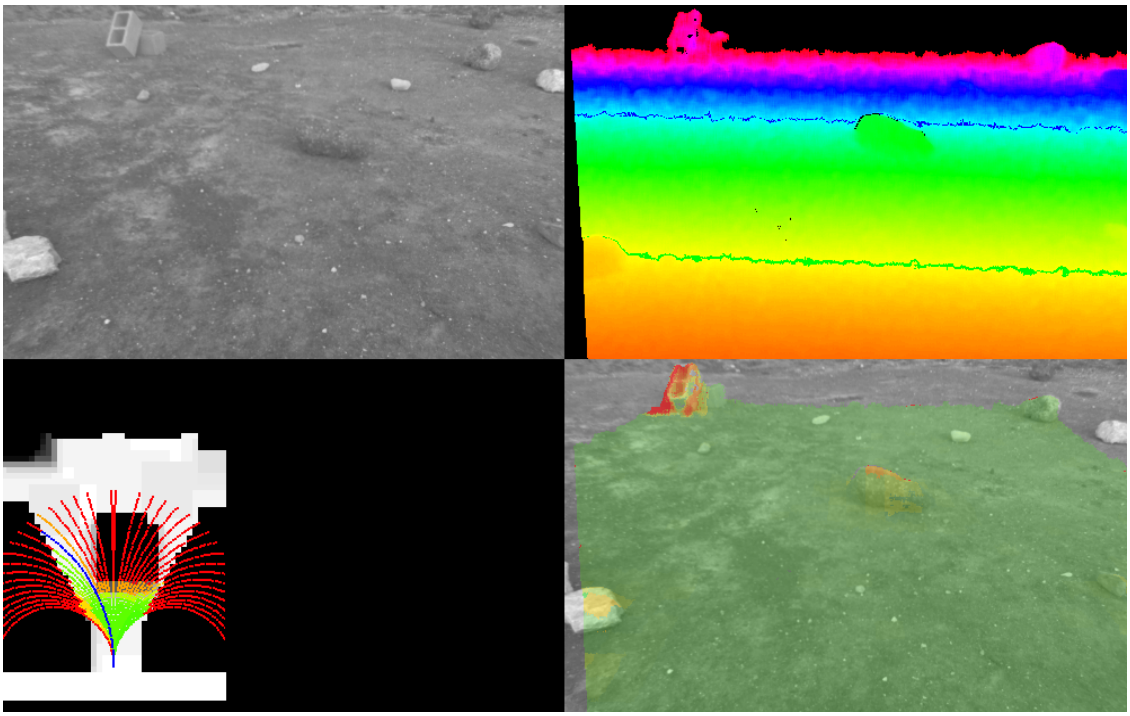


Figure 5: Screenshot from the HFTA-Console. Left to right, bottom to top it is displaying the Grey image, the color coded depth image, the analysis of a pencil of suggested actions (arcs) performed by the obstacle avoidance module in top-down view, and the terrain analysis projected back into the gray image.

Figure 5 shows a screenshot from the HFTA-Console. While the operator is commanding a straight motion, the selected action is a left arc (marked blue), avoiding the rock in front of the robot.

VII. SAFEGUARDED TELEOPERATION EXPERIMENT

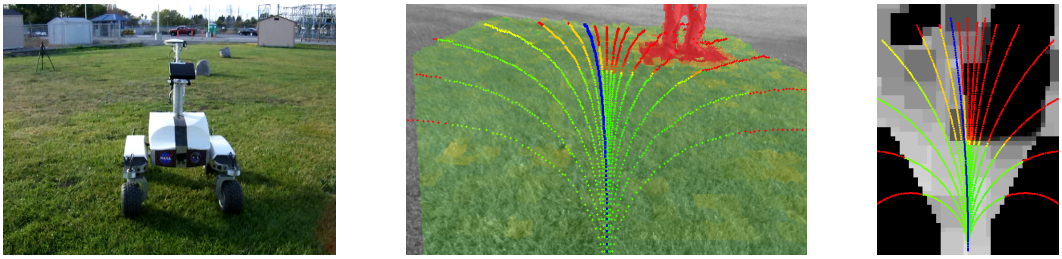
The system was tested and evaluated in the robotic outdoor test facilities at NASA Ames Research Center. The experiments were conducted on a flat lawn surface with artificial rock style obstacles, as well as in the Marscape test area. The former provides a relatively controlled environment for testing specific situations. The later features rocks in various sizes and slopes of differing steepness, exposing the system to a great variety of inputs ranging from safe to hazardous. Rocks of up to 10cm size and slopes up to 20 degrees are considered traversable by the K10 rover. Test runs were documented on video. The complete sensor input was recorded by Miro's logging facilities.[15] The logs were then replayed offline and reevaluated by the HFTA and avoidance module to generate debug visualization with full verbosity.

The robot successfully avoided obstacles (rocks and slopes) while distinguishing traversable small rocks (less the 10cm height) and shallow slopes (less than 20 degrees) from unsafe, untraversable ones. In the remainder of this section, we discuss two experiments from our test-series in more detail.

VII.A. Dynamic Obstacles

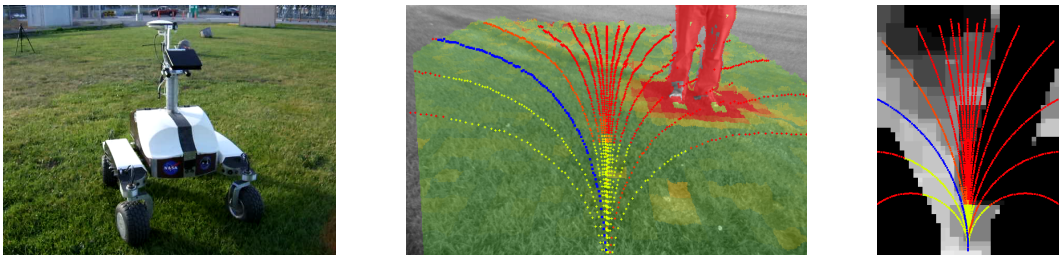
Scenario: Initial experiments verified the base-line obstacle avoidance capabilities with a parcours of static obstacles on the flat lawn. This experiment introduces a dynamic environment. The rover is constantly commanded to drive straight at walking speed. Meanwhile, it is repeatedly interrupted by a human in its path at various distances (see Fig. 6-8 for exemplary imagery).

Target: Safely operating the rover in close proximity to humans was the major goal of this work. This experiment is intended to examine the robot's behavior in proximity of human partners. Even though the obstacle is moving unpredictably, safe avoidance has to be guarantied at any time.



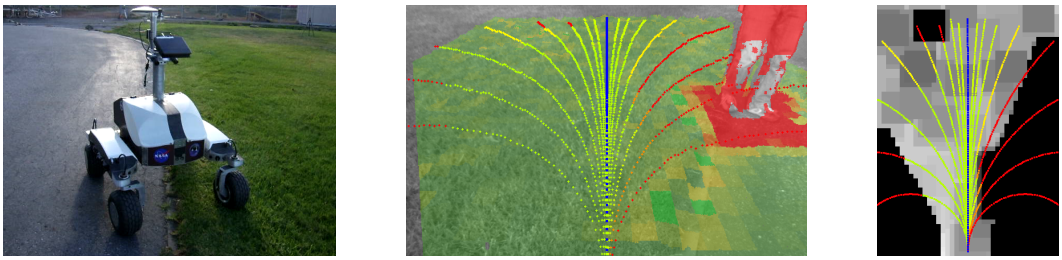
(a) Obstacle view (b) Sensor view (c) Top view

Figure 6: Human obstacle at a distance of 3100mm.



(a) Obstacle view (b) Sensor view (c) Top view

Figure 7: Human obstacle at a distance of 1600mm.



(a) Obstacle view (b) Sensor view (c) Top view

Figure 8: Human obstacle at a distance of 1300mm.

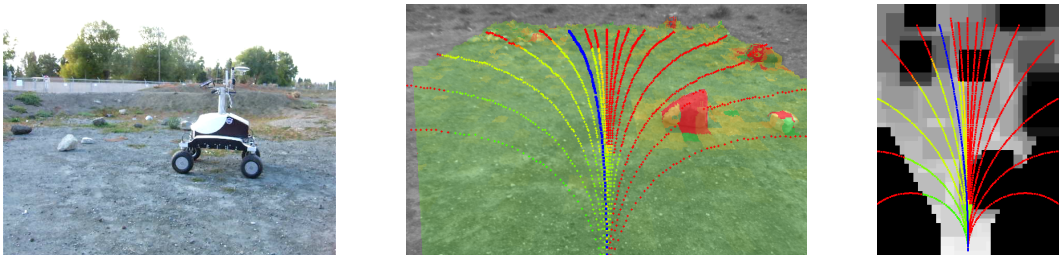
Result: Even though the obstacle was moving, the robot exclusively executed safe motion commands. It always maintained a safety margin to the human and also stopped its movement if no detour was possible. It became apparent though, that the low steering speed does not allow highly dynamic avoidance movements. Interaction could definitely be improved by increasing the robot's steering capabilities.

VII.B. Examples from the Marscape test facility

Scenario: The rover is initially approaching an obstacle field on the Mars-analog testing area Marsscape. It features all characteristics of the targeted environment. The non-vegetated and well textured surface is especially well-suited for a stereo vision based, geometrical terrain traversability analysis system. It contains the expected range of hazardous terrain properties, like rough rocks and steep slopes, in a dense distribution. The user's steering input again had been kept straight.

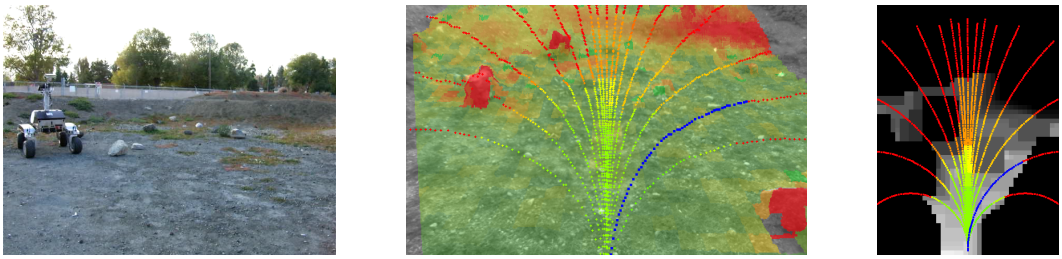
Target: This experiment investigates the overall obstacle avoidance performance of the presented system in a cluttered off-road environment. The dense obstacle distribution presents an advanced challenge. Featuring the full range of obstacle types and sizes, it is intended to confirm the fulfillment of the this work's initial requirements.

Result: Fig. 9-11 show the major challenges of the experiment, which lasted about 20min. During this time, the rover successfully traversed the obstacle field several times and avoided numerous objects. It



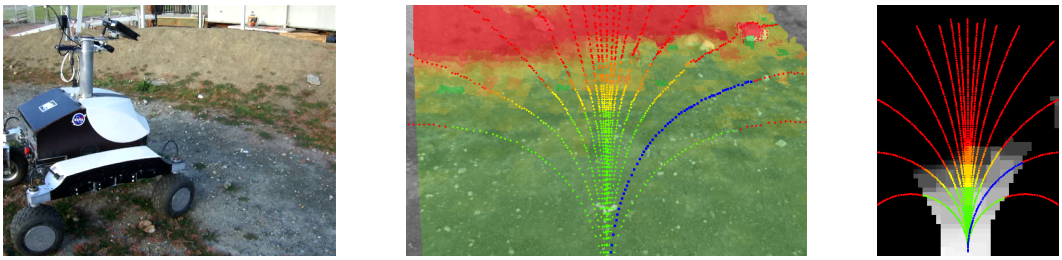
(a) External view (b) Sensor view (c) Top view

Figure 9: Passing the dense obstacle field.



(a) External view (b) Sensor view (c) Top view

Figure 10: Passage into the obstacle field.



(a) External view (b) Sensor view (c) Top view

Figure 11: Robot is facing a steep slope, which poses the danger of tip-over.

successfully distinguished traversable small rocks (less the 100mm height) and shallow slopes (less than 20°) from unsafe, untraversable ones. The avoidance optimality sometimes suffered from the camera's narrow field of view. In rare occasions, the rover got stuck in a dead end because an exit was not in view. An additional behavior, which would have rotated the rover in place to examine the complete surrounding, could have resolved this issue.

VIII. CONCLUSIONS AND FUTURE WORKS

The test scenarios for the full control-loop on-board the K10 rover have been a great success, as the concept's applicability could be confirmed experimentally. The rover had been safeguarded by the avoidance system at all times. The high update frequency of the control loop make the system applicable for robots moving at walking speed in the proximity of human crew.

A shortcoming of the tested setup was the low steering speed of the K10 rover. Its actions appeared sluggish when it came to close interaction. Many stopping maneuvers occurred in situations where uninterrupted avoidance by quick steering might have been possible. This behavior was intensified by the narrow field of view of the stereo camera. The obstacle map was often completely occluded by objects which were especially close to the rover. Potential escape routes were not visible to the avoidance system. Also, based on our tests, some improvements need to be made to improve stereo performance in high-contrast environments. In particular, avoiding imager saturation under bright lighting is a priority for use in outdoor applications.

Apart from that, the Tyzx G2 showed to be an ideal sensor solution for that purpose. Off-loading stereo reconstruction from the robot's CPU and providing range images at 30Hz was an important improvement over the previous configuration. The reasonable noise characteristics in the target distance of approximately 1000-5500mm eased measurement analysis.

Future work will try to overcome the limitations documented under testing. The steering speed will be improved and we will evaluate the possibilities of mounting a second 3D sensor, to double the field of view. Additionally, we are investigating implementing parts of the terrain analysis algorithms on the stereo-processing hardware, further reducing computational requirements on the host system. For an extensive evaluation, we plan to field-test the system in one of the upcoming lunar-analog robotic tests of NASA's Human Robotic Systems project.

ACKNOWLEDGMENTS

We would like to thank CMU West for hosting Thomas as a visiting researcher at Ames Research Center, and Tyzx Inc. for their support with the G2 system. This work was partially funded by the ETDP Human Robotic Systems project.

References

[1]Park, E., Kobayashi, L., and Lee, S., "Extensible Hardware Architecture for Mobile Robots." *ICRA*, IEEE, IEEE, Barcelona, Spain, April 2005, pp. 3084–3089.

[2]Woodfill, J. I., Gordon, G., Jurasek, D., Brown, T., and Buck, R., "The Tyzx DeepSea G2 Vision System, ATaskable, Embedded Stereo Camera," *Embedded Computer Vision*, 2006, p. 126.

[3]Fong, T. W., Nourbakhsh, I., Ambrose, R., Simmons, R., Schultz, A., and Scholtz, J., "The Peer-to-Peer Human-Robot Interaction Project," *AIAA Space 2005*, September 2005, AIAA-2005-6750.

[4]Smith, P. H., Bell, J. F. r., Bridges, N. T., Britt, D. T., Gaddis, L., Greeley, R., Keller, H. U., Herkenhoff, K. E., Jaumann, R., Johnson, J. R., Kirk, R. L., Lemmon, M., Maki, J. N., Malin, M. C., Murchie, S. L., Oberst, J., Parker, T. J., Reid, R. J., Sablotny, R., Soderblom, L. A., Stoker, C., Sullivan, R., Thomas, N., Tomasko, M. G., and Wegryn, E., "Results from the Mars Pathfinder camera." *Science.*, Vol. 278, 1997.

[5]Goldberg, S., Maimone, M., and Matthies, L., "Stereo vision and rover navigation software for planetary exploration," *Aerospace Conference Proceedings, 2002. IEEE*, Vol. 5, 2002, pp. 5–2025–5–2036 vol.5.

[6]Volpe, R., "Rover technology development and mission infusion beyond MER," *Aerospace Conference, 2005 IEEE*, 2005, pp. 971–981.

[7]Urmson, C. P., Dias, M. B., and Simmons, R. G., "Stereo Vision Based Navigation for Sun-Synchronous Exploration," July 11 2002.

[8]Ye, C. and Borenstein, J., "T-transformation: traversability analysis for navigation on rugged terrain," *Unmanned Ground Vehicle Technology VI. Edited by Gerhart, Grant R.; Shoemaker, Chuck M.; Gage, Douglas W. Proceedings of the SPIE, Volume 5422, pp. 473-483 (2004).*, edited by G. R. Gerhart, C. M. Shoemaker, and D. W. Gage, Vol. 5422 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, Sept. 2004, pp. 473–483.

[9]Simmons, R., Henriksen, L., Chrisman, L., and Whelan, G., "Obstacle Avoidance and Safeguarding for a Lunar Rover," July 19 2001.

[10]Fong, T. W., Deans, M., Bualat, M., Flueckiger, L., Allan, M., Utz, H., Lee, S., To, V., and Lee, P., "Analog Lunar Robotic Site Survey at Haughton Crater," *Workshop on Enabling Exploration: The Lunar Outpost and Beyond*, No. 3058, LEAG, October 2007.

[11]Singh, S., Simmons, R. G., Smith, T., Stentz, A., Verma, V., Yahja, A., and Schwehr, K., "Recent Progress in Local and Global Traversability for Planetary Rovers," *ICRA*, IEEE, IEEE, San Francisco, April 2000, pp. 1194–1200.

[12]Hamilton, J., *Path Planning for Rover Navigation*, Master's thesis, EPFL, September 2007.

[13]Flueckiger, L., To, V., and Utz, H., "Service Oriented Robotic Architecture supporting a Lunar Analog Test," *Proceedings of the 9th Symposium on Artificial Intelligence, Robotics, and Automation in Space (iSAIRAS 2008)*, Los Angeles, California, February 2008.

[14]Utz, H., Sablatnög, S., Enderle, S., and Kraetzschmar, G. K., "Miro – Middleware for Mobile Robot Applications," *IEEE Transactions on Robotics and Automation, Special Issue on Object-Oriented Distributed Control Architectures*, Vol. 18, No. 4, August 2002, pp. 493–497.

[15]Utz, H., Mayer, G., and Kraetzschmar, G. K., "Middleware Logging Facilities for Experimentation and Evaluation in Robotics," 27th German Conference on Artificial Intelligence (KI2004), September 2004, Workshop on Methods and Technology for Empirical Evaluation of Multiagent Systems and Multirobot Teams.