2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

High Fidelity Day/Night Stereo Mapping with Vegetation and Negative Obstacle Detection for Vision-in-the-Loop Walking

Max Bajracharya[1], Jeremy Ma[1], Matt Malchano[2], Alex Perkins[2], Alfred A. Rizzi[2], Larry Matthies[1]

Abstract— **This paper describes the stereo vision near-field terrain mapping system used by the Legged Squad Support System (LS3) quadruped vehicle to automatically adjust its gait in complex natural terrain. The mapping system achieves high robustness with a combination of stereo model-based outlier rejection and spatial and temporal filtering, enabled by a unique hybrid 2D/3D data structure. Classification of sparse structures allows the vehicle to traverse through vegetation. Inference of negative obstacles allows the vehicle to avoid steep drop-offs. A custom designed near-infrared illumination system enables operation at night. The mapping system has been tested extensively with controlled experiments and 72km of field testing in a wide variety of terrains and conditions.**

I. INTRODUCTION

Terrain mapping from stereo and LIDAR sensors is commonly used for autonomous navigation of wheeled vehicles. Typically, these terrain maps are sufficient to identify obstacles and compute a cost of traversable paths for use in vehicle motion planning. However, for highly capable vehicles that can traverse complex terrain, such as steep, rocky slopes, or dense vegetation, the resolution and accuracy of maps generated using existing techniques is not sufficient. Using a sensor-based terrain map to adjust the desired foot placement of a legged robot requires mapping with a fidelity that is higher than the foot size. Furthermore, terrain that can or cannot be stepped over or through must be classified.

We have developed a stereo-vision-based terrain mapping system that enables closed-loop gait adjustment of a dynamic walking robot. The mapping system is used during nominal operations by the Boston Dynamics Legged Squad Support System (LS3) quadruped vehicle (Figure I), developed under the DARPA LS3 program. While the LS3 vehicle is highly capable and robust without any vision feedback, the map produced by the stereo vision system enables the vehicle to predict the need for gait adjustments, leading to even higher levels of robustness. In addition to the classification of terrain step height, detection of sparse vegetation enables the robot to traverse through terrain that would otherwise appear as obstacles. Similarly, detection of lethal negative obstacles could allow the vehicle to avoid areas with very steep slopes. The mapping system operates with no modifications during night-time operations by using near-infrared (NIR) illuminators. The illumination system is designed as a set of independently controllable illuminators configured to provide uniform illumination over the stereo camera field-of-view.

Our near-field mapping system is designed to achieve a terrain map fidelity of better than 5cm (1/3 of the 15cm foot size) out to 5m. Stereo cameras are used because they are the only sensor that can provide the angular resolution required for producing a map of this fidelity within the system's size, weight, and power constraints. They are advantageous because they produce a full frame of range data, and can be acquired at high rate. On the other hand, dense stereo adds computation time and produces noisy data and artifacts in the range data. When used for vision in-the-loop adjustment of gaits, inaccuracies in the map are unacceptable because they can directly cause the vehicle to destabilize or fall. The mapping system uses a unique hybrid 2D/3D data structure and stereo modeling to enable computationally efficient methods of rejecting outliers and temporally and spatially filtering the range data to produce robust and accurate maps. The map is effective in dynamic environments, including moving objects and conditions such as rain and snow, and handles degraded data, such as at night or when walking through tall vegetation.

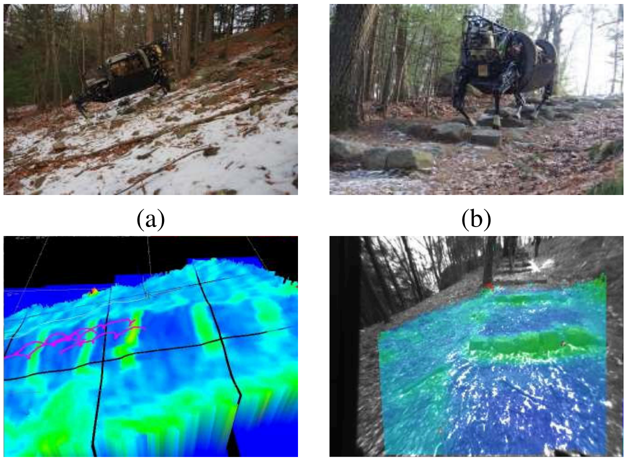


Fig. 1. (a) and (b) show the LS3 vehicle walking up and down a set of stairs on a forest trail; (c) shows the 2D terrain map color coded by step height (from flat (blue) to moderate (green) to high (red)), with the foot traces in magenta; and (d) shows the backprojection of the map overlaid on the left stereo image.

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology and Boston Dynamics. The work is funded by the DARPA LS3 program. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

[1] Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr., Pasadena, CA 91109, maxb@jpl.nasa.gov

[2] Boston Dynamics, 78 Fourth Ave., Waltham, MA 02451

978-1-4673-6358-7/13/$31.00 ©2013 IEEE

A. Related Work

Occupancy grids have historically been the most common representation for terrain maps [1], [2]. They have been successfully used for autonomous wheeled vehicles operating in urban [3], cross-country [4], [5], and indoor [6] environments. For outdoor environments, a 2D elevation map is typically used, while for indoor or more complex environments, a 3D voxel map is used. For efficiency, most 3D maps only store occupancy information [6], [7] and have mainly used LIDAR data with spatial filtering [8] and ray constraints for eliminating dynamic obstacles. As a result, these maps are typically too low resolution for computing footstep locations. LIDAR and stereo have been used for building maps for legged locomotion [9], [10], [11], [12], but none have been tested or fielded to the extent of the approach described here.

More recently, structured light depth sensors have been used to build accurate local [13] and extended [14] maps of registered point clouds. These approaches have been focused on producing visually realistic models, but do not result in a map that can be used for efficient motion planning. Furthermore, these methods do not address the noise and artifacts caused by stereo, which is typically much less for structured light systems. Finally, structured light systems can currently only be used indoors or in low-light environments.

Filtering noisy stereo data to produce higher fidelity maps has been performed in a standard height map [15], as well as in a representation that accounts for stereo range error [16]. This has been extended to 3D representations such as occupancy maps [17], with data structures that allow efficient nearest neighbor queries [18], or in other similar representations [19]. However, most of these approaches are considerably slower than the approach described here and do not handle dynamic environments or outliers. Another way to improve stereo mapping is to improve the stereo matching algorithm itself. While there are many techniques for doing this, none are fast enough for use on a real-time system.

Detecting negative obstacles in maps has been performed using image space techniques, thermal imagery [20], LIDAR and map classification [21], and ray tracing [22] methods. The latter is similar to our approach, although we build the map first and walk along the 2D map rather than explicitly tracing rays, making it significantly more efficient. Vegetation detection has typically focused on the multi-spectral signature of green vegetation [23], [24] and the geometry from LIDAR. We do not explicitly detect green vegetation, but rather detect sparse structures. This works on green and non-green vegetation, but could be confused by structures such as chain link fence.

II. APPROACH

Our mapping system produces a high fidelity terrain map that includes the surface elevation, local statistics such as slope and roughness, and discrete classes such as positive and negative obstacles. The system can operate in dynamic environments and labels and tracks moving (positive) obstacles, while properly mapping terrain over where the objects travel. The output of the system is represented as a 2D structure of map cells, tiled into blocks that are sent to the planning and control modules, which are running concurrently on the robot. Each cell contains the geometry statistics and classification labels.

A standard dense local correlation stereo matching algorithm [25] is used to produce the input range data for the system. The vehicle pose is estimated accurately using visual odometry combined with an IMU and leg odometry in a Kalman filter framework [26]. A hybrid 2D/3D structure of voxels is populated with the dense stereo range data. The voxels are filtered temporally and vertically to reject geometric artifacts common in stereo data. The resulting map of ground elevations, statistics, and classifications is used to adjust parameters of the vehicles movement on approaching terrain, such as lifting the feet higher to avoid an obstacle. Vegetation classification is used to label locations in the map as areas containing sparse geometry. Negative obstacles, such as ditches, are detected and marked. Finally, a custom NIR illumination system is used to enable range-limited night operation with no change of software or parameters.

A. Data Structure

The mapping system outputs a 2D map of cells to the vehicle control module. Internally, the mapping system maintains a representation using a hybrid framework of dense 2D and sparse 3D voxels. The 3D representation primarily handles dynamic, complex environments and the inherently noisy stereo data, while the 2D representation efficiently computes local statistics.

The map data structure is comprised of a set of tiles. Each tile contains a 2D array of voxel column pointers, with each column bounded to a fixed height above and below the tile origin, as well as 2D statistics, accumulated and then filtered for each cell. A fixed number of voxels and tiles are allocated onto a local heap and used and returned as necessary. When range data is first projected into a tile and voxel, the pointers are populated with a new tile and/or voxel from the heap. The tile origin is set to an estimated ground plane below the vehicle (based on the foot location). Because the map is only used for terrain classification, any data that projects above the vehicle height is not considered, allowing the data structure to be kept as a single layer of tiles. Once a voxel is allocated, any new range data that falls into that voxel is added, and sums of first and second order statistics are maintained, and the voxel and tile are marked as “dirty”.

B. Error Modeling and Filtering

The key to being able to use a stereo-vision map in a closed-loop system that adjusts gait parameters and footsteps is producing a map that is geometrically accurate and free of false positives and artifacts. Typically, spatial and temporal filtering is used to abate noise in stereo range data. However, we have found it necessary to also use explicit outlier rejection based on stereo error modeling to produce very high quality maps.

After projecting the range data for a single stereo frame into voxels, the mapping system collapses all dirty voxel columns in the tiles to a single 2D tile cell, rejecting spatial and temporal outliers. After the initial projection into a voxel column, the algorithm can estimate the ground surface elevation of that column. Once this elevation is known, individual fixed-height floating voxels are no longer needed. A minimum number of projected stereo ranges must be reached by a given voxel to be valid. The standard stereo error model [27] is used to set this minimum number to be the expected point count on a virtual ground plane at the range of each voxel. The model is conservative because a vertical obstacle will have significantly more hits than the ground plane due to the angle of incidence. This technique is effective in eliminating the "streamer" or "spike" artifacts commonly seen in stereo data. Otherwise, small numbers of overhanging range points contaminate the ground surface estimation, particularly at longer range where fewer points are already available.

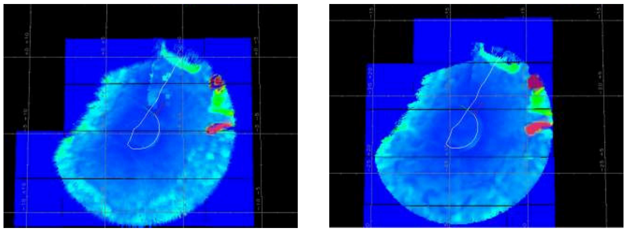

(a) (b)

Fig. 2. (a) shows the map color coded by step height without age-weighted filter and outlier rejection, which results in artifacts (lighter regions) due to spatial filter; (b) shows the same map with age-weighted filtering and outlier rejection, which eliminates the artifacts.

If stereo range data from the current frame projects to a voxel, that voxel is used in maps containing its associated column. If a voxel is not observed in the current stereo frame, then based on its last observation time, its total observations, and the number of observations in its associated column, it is rejected as a temporal outlier. Voxels, both used and unused, are kept so that data may accumulate over time. Because of the stereo model-based outlier rejection, this leads to more aggressive elimination of higher variance data that was observed in the far-field, which is replaced by more accurate data in the near-field. It also allows dynamic objects to be eliminated after they move.

After the voxel data is accumulated into a 2D representation, the dirty tile cells are spatially filtered with an age-weighted 2D separable, running sum filter. Using weights that are a function of age allows the filter to be applied over regions where there is overlap between new and old data that may be inconsistent due to pose drift. Figure 2 illustrates the difference between standard and age-weighted filtering.

While the technique is capable of eliminating dynamic obstacles from the scene, it takes at least one frame to do so. The vehicle can occasionally see its own feet, which results in a single frame of false positive obstacles (the knees are not a problem because they always fall nearer than the minimum stereo range). In this case, because the feet positions are known, they are explicitly removed before projecting data into the map by masking out the range data in the image (Figure 3).

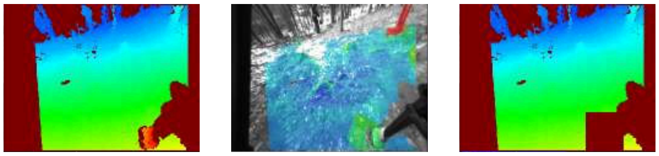

(a) (b) (c)

Fig. 3. (a) Shows the stereo disparity on an image where the foot enters the field of view, and causes a false obstacle detection, as seen in (b) which shows the back projected map; (c) shows the stereo disparity with the foot being masked out.

C. Gait Adjustment

As a vehicle operates over changing terrain, parameters controlling the way it moves (i.e. its gait) need to change. For example, legged vehicles may elect to lift their feet higher to step over trip hazards, adjust their pose to match terrain shape, or speed up or slow down to navigate upcoming terrain. The robustness and accuracy of the stereo map enables it to be used to continuously adjust the parameters of the vehicle's gait. While the vehicle already has the ability to proprioceptively sense rough or sloped terrain and adjust its gait in response to the terrain, using vision "in-the-loop" allows the vehicle to predict the need to adjust its gait. This allows the vehicle to avoid disturbances caused by obstacles, rather than needing to recover from them.

The primary parameter that is automatically adjusted based on the map is the gait's swing height, which is the peak height above the ground that a foot reaches while moving between footholds. This can be calculated for the gait as a whole, or for each foot individually. One approach to computing an overall gait height is to compute roughness statistics over a region in front of the robot. Alternatively, by examining the map cells along the path of each leg's swing trajectory, the swing height can be adjusted for each leg as a function of the maximum expected step height. Another approach would be to adjust the target footstep location based on the map data. However, this approach reduces the system's natural ability to react to disturbances, and remains an area of ongoing research.

Of the multiple approaches considered, we found that adjusting only the swing height of each leg independently produced the most robust and effective results. Adjusting the swing height of all the legs when rough terrain is detected results in excessive motion and unnecessary efficiency loss. Simply adjusting the swing height of each leg independently increases robustness to obstacles, while maintaining the overall stability and efficiency of the vehicle. As a result, during normal operation, only the individual leg swing height is adjusted.

D. Vegetation Detection

With a highly capable legged vehicle, elevation geometry statistics alone can be insufficient to adjust its gait parameters. For example, when walking in ankle or knee high grass, the vehicle should not step as high as it might when walking through a rocky field or over steps of similar height variance. To address these situations, we added a classification mode to the mapping system that classifies terrain that does not present a hazard to the robot, but has similar statistics to something that would.

Our key observation is that obstacles that do not present a hazard to the vehicle tend to be thin structures, such as grass or sparse bushes, as opposed to trees or rocks. These objects tend to have a large number of depth discontinuities or poor, but non-zero stereo coverage. Furthermore, they typically also have high texture, implying that they should have dense stereo coverage. We exploit this phenomenon in a computationally efficient manner by processing the disparity and rectified image directly, and then projecting occurrences into the map to accumulate their statistics. Note that we do not explicitly detect vegetation, but rather collections of thin, sparse structures.

First a simple and fast texture measure is computed using a running-sum filter of horizontal pixel differences over the size of the stereo correlation window. The average sum over all pixels that have stereo data is kept as well. Then each pixel that is not saturated and has a texture value greater than the average is considered. For each of these pixels, if the horizontal and vertical neighbor pixels are a fixed range away, the central pixel is then only classified as a depth discontinuity if any of these range deltas is beyond a fixed value, or has no disparity. Only data that is within a fixed range of the vehicle is processed, as stereo error would make the range test invalid at longer range, and a delta disparity check is less informative.

The depth discontinuity count is projected into the map and accumulated both temporally and spatially in the map. The accumulated value can be thought of as a crude approximation of an object's sparseness or density. Then a simple classifier using the discontinuity count and the geometry statistics can be used to determine if the filtered map cell should be considered a reliable and solid obstacle or not.

E. Negative Obstacle Detection

When operating near large negative obstacles, the vehicle can become stuck, fall large distances, or entrap its legs. Detection allows the vehicle to avoid these hazards. However, negative obstacles must be distinguished from traversable downslopes and visually confounding phenomena.

For stereo systems, negative obstacle detection can be performed in the image or map. Higher fidelity detection requires walking rows and running a classifier in the image, but can be prone to false positives due to stereo noise. Since we are only interested in detecting relatively large lethal negative obstacles with high reliability, and computation time is critical, we use an efficient map-based approach.

After each stereo frame is projected into the map and the 2D filtered map is computed, we walk 2D rays from the projection of the camera to the ground plane, out within the bounds of the camera field of view. For each ray walked, an existing ground surface cell indicates a minimum slope (or maximum elevation) beyond that cell based on the ray from the camera. Cells that do not have data and that are not saturated are then labeled with a maximum elevation based on the ray. For empty cells, the next valid cell along the ray (if there is one) provides an indication of the minimum elevation. If there is no valid cell beyond the missing data, the ray extended out to a fixed distance is used to compute the minimum elevation. The difference between the maximum and minimum elevation is then used as an estimate of the negative step height.

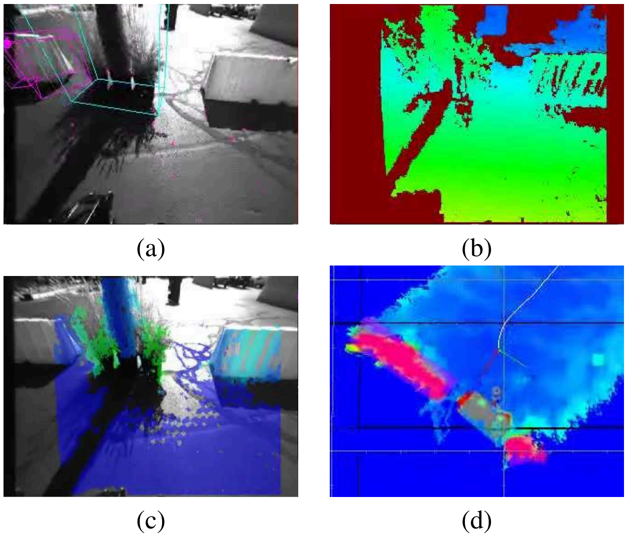


Fig. 4. (a) Shows the stereo image for a controlled experiment for vegetation detection; the cyan boxes are sparse detected obstacles, while the magenta boxes are dense detected obstacles; (b) shows the stereo disparity; (c) shows the continuous classification of sparse vs. dense obstacles; and (d) shows accumulated map with cyan/magenta boxes for sparse/dense obstacles.

To make computing the negative step height efficient, portions of the ray that overlap with an already processed ray are reused. Because the ray is being traced in a 2D structure of fixed size cells, rays near the robot will be reused more often and save considerable computation. While not strictly correct since a ray hitting a cell may have approached it from a slightly different angle, the approximation is sufficient for computing coarse geometry. Similarly, for each ray the next valid cell along the ray is only computed as necessary, and reused along the ray until it needs to be recomputed, saving significant computation time.

Because negative obstacles are identified based on the terrain geometry, regions behind tall obstacles will have a negative step height. Furthermore, since the vehicle can operate in high slope environments, negative obstacles can exist well above the vehicle's feet (i.e. the back side of a sloped surface). Consequently, to eliminate negative obstacles from behind positive obstacles, we explicitly detect a fixed size positive obstacle and do not label any negative obstacles for rays that pass over the obstacle. Finally, to eliminate small false positive negative obstacles, which are considered lethal and cause incorrect deviations in the planner, we apply an angular filter to adjacent negative obstacle cells. This requires that negative obstacles span a fixed minimum angle (of 5 degrees), as seen by the camera. In particular, this filter helps to remove artifacts on the edge of the field-of-view, where a valid ray exists, but the cells have not been observed many times yet (as the robot is rotating) and so are all invalid.

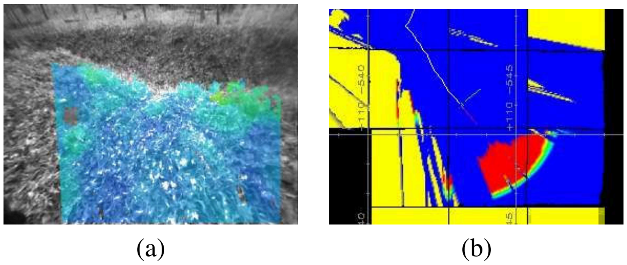

(a) (b)

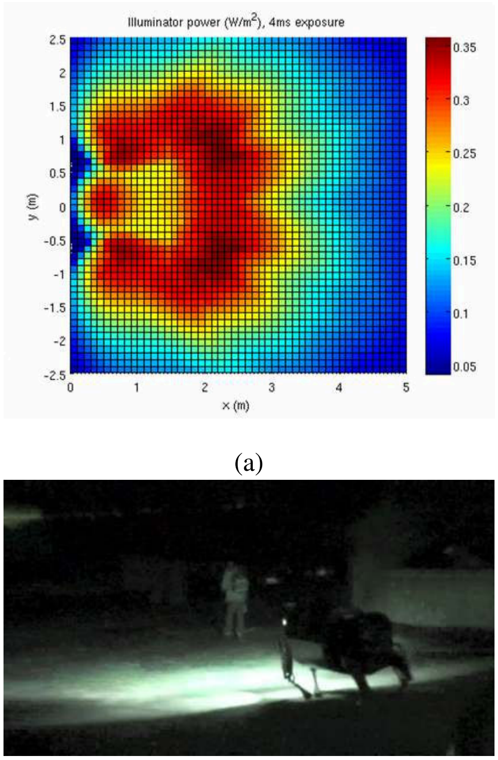
Fig. 5. (a) Shows the stereo image of a large ditch, with the map overlaid, and (b) shows the detection of the negative obstacle (blue to red is the severity of negative obstacle, and yellow are regions that are not classified).

F. Illuminator Design

The Legged Squad Support System vehicle has a requirement to operate during both the day and night. While thermal infrared cameras were considered for near field mapping, visible cameras were chosen for resolution, cost, and performance reasons. To operate at night, near-infrared (NIR) LEDs are used to illuminate the terrain in front of the robot, and the mapping algorithms are used unchanged.

While the visible camera CCDs used have reasonable NIR sensitivity, it is still approximately 1/10th the sensitivity of that in the visible spectrum. Coupled with the requirement to see from 1 meter to 5 meters away, the camera does not have the dynamic range to operate with a single uniform light source. Consequently, we designed a tiled illuminator system with multiple light power levels and lenses to achieve uniform lighting at varying distance. The power level of the individual illuminators are also independently controllable, which means that as the vehicle approaches a large obstacle, the illuminators can be used to prevent saturation in the image.

We used a model-based approach to designing the illuminator system. We collected illumination data for several combinations of NIR illuminators and lenses, which we then used to compute an illumination model for each one. We then computed the expected illumination pattern on a flat ground plane for combinations of different illuminators and pointing angles (Figure 6). To balance performance and system complexity, we ultimately selected a single LED type, used in a two row pattern, with one lens type for all the far-field illuminators (top row), and another lens type for all the near-field illuminators (bottom row). The final design uses 10 8 Watt NIR (810nm) LEDs with 30 FOV and 60 FOV type lenses. The LEDs are triggered along with the cameras and so only operate when the cameras are exposing, and are limited to a 10ms exposure time. They are PWM throttled to control the total illumination output level. This not only reduces the total power required, but makes the illuminators eye safe at approximately 0.3m. Operational protocols are used to ensure people keep the required distance when the illuminators are on.



(a)

(b)

Fig. 6. (a) shows the predicted model of the illuminator design on flat ground; (b) shows the LS3 vehicle with the illuminators on at night, shot with a night vision camera.

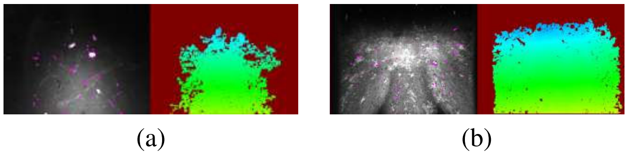

(a) (b)

Fig. 7. (a) Shows the stereo coverage of a uniform illuminator design; (b) shows the stereo coverage of our new, multi-illuminator design.

III. EXPERIMENTAL RESULTS

The basic mapping system described here has been fielded in incremental fashion as part of the DARPA LS3 program, starting in 11/2010 on the BigDog vehicle, and then since 1/2012 on the LS3 vehicle. Since early in the program, the map was used for basic robot level path planning and obstacle avoidance. Even when the vehicle was not operating in a mode that required the map, the map was being generated. However, only with the use of the filtering and outlier rejection techniques recently developed and described in this paper, has the map been robust enough to be used in closed-loop fashion to adjust the vehicle's gait. The vegetation and negative obstacle detection are still experimental features and have only been tested in controlled experiments.

On the LS3 vehicle, during 9 official field tests, over 23 different days, and 245 distinct runs, the mapping system has seen approximately 28 hours of data, covering 72km, and 620K frames. This does not include any controlled testing or experiments not performed during official field testing. The system is running the mapping system while running under normal operation, and all input (imagery, IMU, etc.) and output (pose, maps, etc.) data is logged. In total, this consists of approximately 5TB of data, dominated primarily by (uncompressed) imagery data.

Quantitatively evaluating the mapping performance in natural terrain is difficult. We have used simulated imagery of known scenes to evaluate the accuracy of the map in simple scenarios, but this does not reflect real-world scenarios well. As a result, we tend to initially evaluate the mapping performance in controlled experiments, with known objects in specific configurations, and then test the system extensively in the field under natural conditions. After every field test is complete, the log data of every run is used to visually inspect the map quality. The planning team reports any case where the plan or robot acts in a way it shouldn't, and the map is inspected at that point to understand why the decision was made. Similarly, the controls team reports when the gait is adjusted from nominal, and reports cases that appear incorrect.

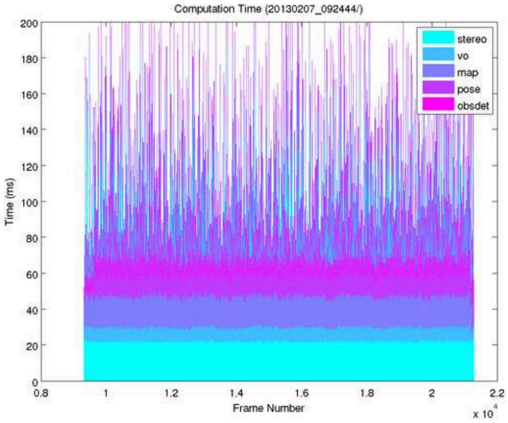
A. Computation Time

The average computation time for a stereo frame for a typical run in a forest environment is shown in Figure 8. The stereo and vegetation detection processes are constant time; visual odometry and negative obstacle detection vary only slightly as a function of the environment; and mapping varies as a function of vehicle speed and the amount of 3D structure in the scene. For the run shown, dynamic obstacle detection and tracking is disabled.

B. Controlled Experiments

Controlled experiments in a parking lot augmented with different terrains are used to validate the system before field tests. The standard test includes urban positive obstacles, such as trash cans and cars, potted plants, steps and logs, a rock field, and a grassy sloped hill. The quality of the map is validated offline in two ways. The first method compares the quality of the map generated online to one that is generated offline using stereo computed at a higher resolution, and only in the very near-field. Local statistics of the maps are computed for regions under the robot and compared. For example, Figure 9 (a) shows the difference in elevation variance computed with full resolution (1024x768) stereo data, out to varying ranges, with that computed at the resolution used onboard (512x384) without using model-based filtering. This illustrates how noisy far field data can corrupt the map. When using the model-based filtering, the elevation variance of the full 5m map is the same quality as the 2m map, for data out to 2m.

The second method of evaluating the map quality involves manually ground truthing the times at which the robot enters and exists different terrain conditions, such as entering the rock field, or stepping over a log. The map statistics during the different terrain conditions are then examined for discernability. Figure 9 (b) shows the elevation variance under the robot using model-based filtering for a typical run, showing hills, steps, and a rock field, with asphalt in between. Since the map is used to adjust the gait swing height on these terrains, they must be distinguishable from the nominal conditions. Without the model-based filtering, the elevation variance over the asphalt is significantly noisier, and difficult to distinguish from the rock field.



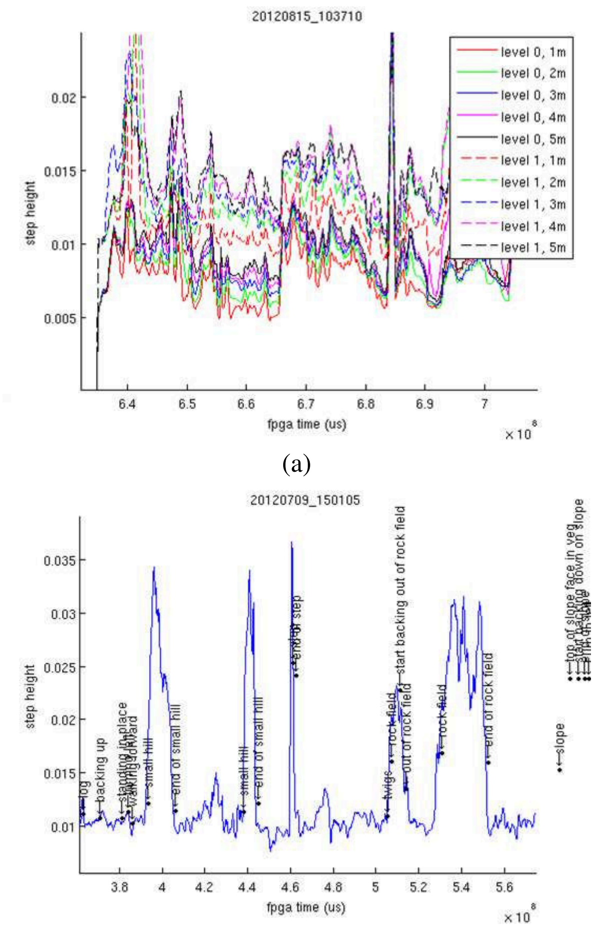
Algorithm	Avg. Time
Stereo	31ms
Visual Odometry	9ms
Mapping	19ms

Fig. 8. The computation time for a typical run through a forest trail; the system does not run on a real-time operating system, which results in the spikes in processing time. The system runs on a 2.66GHz Intel Core i7 CPU.

Specific terrain conditions are tested by creating obstacles that simulate the real scenarios. For example, to test vegetation detection, sparse potted plants are placed in between a set of barriers and the vehicle is commanded to a goal location on the opposite side. When vegetation detection is enabled, the robot is expected to walk through the vegetation. Similarly, a ramp leading to a drop-off was constructed to test negative obstacle detection.

C. Field Tests

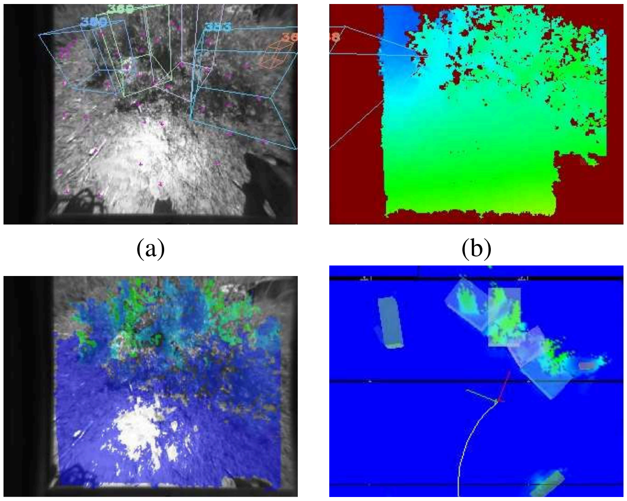
Throughout the LS3 program, field tests are being conducted approximately every 6 weeks. Earlier in the program, field tests were conducted locally at various sites in the northeast region of the United States. These included mostly forest environments, with and without vegetation, and of varying density of rocks, trees, mud, puddles, and slopes. Tests are conducted both on trails and roads of varying size, as well as cross-country. Tests are also conducted in almost any weather, including rain and snow. A paintball site was also used to test operation in more urban environments. Table I summarizes the locations, distances, and run time of the various sites used for testing.



(a)

(b)

Fig. 9. (a) Shows the map elevation variance for full and half stereo resolution, with data out to varying maximum distances projected into the map, with no outlier rejection; (b) shows the elevation variance features computed under the robot for a typical run with ground-truth labels for different terrains.



(a)

(b)

(c)

(d)

Fig. 10. An example of vegetation detection in natural conditions; (a) shows the stereo image with cyan boxes representing sparse obstacles; (b) the stereo disparity; (c) the back projected map with blue to green indicating sparseness; (d) the map, with blue to gree indicating spareness and cyan boxes around sparse obstacles.

Location	Time	Distance
Forest (MA, winter)	3.2hrs	10.4km
Forest (MA, spring)	3.7hrs	4.7km
Vegetated fields/trails (MA, summer)	1.8hrs	4.9km
Paintball site (MA, summer)	0.5hrs	1.0km
Vegetated fields/trails (NH, fall)	1.1hrs	3.5km
Paintball site (MA, fall)	1.2hrs	2.3km
Vegetated fields/trails (NH, winter)	1.2hrs	4.2km
Paintball site (MA, winter)	1.3hrs	2.9km
Sparse/dense forest, fields (VA, winter)	15.0hrs	39.6km
Total:	27.9hrs	73.5km

TABLE I

LIST OF SITES WHERE THE LS3 MAPPING SYSTEM HAS BEEN TESTED.

While field tests are mostly used to assess performance of the system over typical operating conditions, they also serve to collect data of more difficult terrain and test specific terrain cases. For example, Figure I shows a specific run in a park in Waltham, MA, where the vehicle was following a leader up a makeshift staircase on a trail. Like all other runs during this field test, the stereo map was being used to automatically adjust the gait parameters, but it provided a run in which analyzing the impact of the vision system was easy. On this run, of the 368 steps taken, 28 were automatically adjusted by the vision system. This same run has been performed without the vision system, but required manual adjustment of the robot's step height which in turn reduced the robot's efficiency.

As the LS3 program continues and the vehicle matures, the vehicle is being tested in longer (generally two week) tests throughout the United States. The first of these tests was conducted at Fort Pickett, VA and included sparse and dense forests with fallen logs (Figure 12), fields with tall and short grass and other vegetation, long gravel and dirt roads, day and night operations, and an urban MOUT site (Figure 11). During this test, the mapping system was run over 40km, with the vehicle walking an average speed of 1m/s. The vehicle successfully navigated most of the terrain.

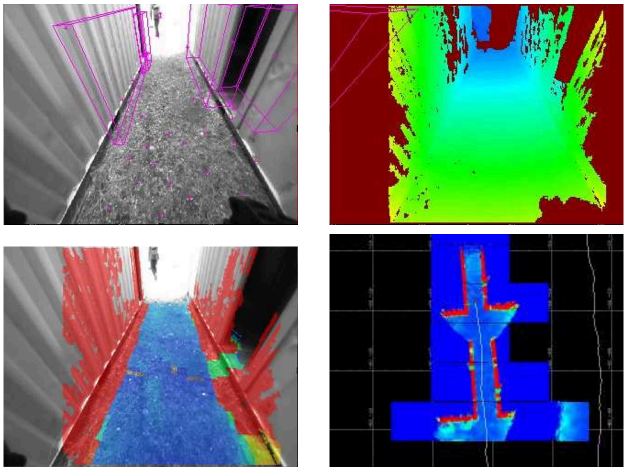


Fig. 11. An example of the terrain mapping system being used in a MOUT site

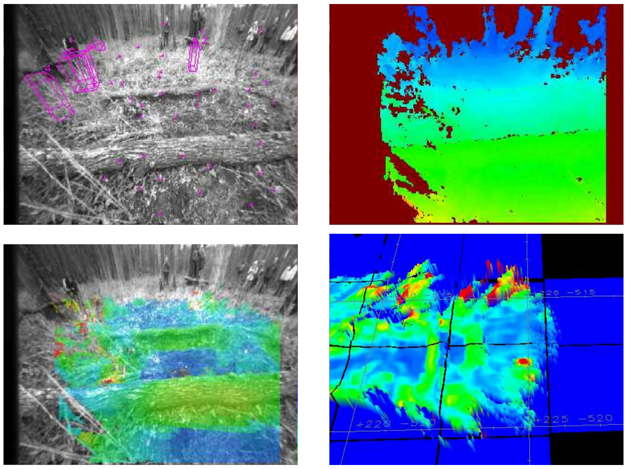


Fig. 12. An example of the terrain mapping system being used in a forest with fallen logs.

IV. CONCLUSIONS

We have developed a stereo vision system capable of mapping natural terrain to the fidelity required for automatically adjusting a legged vehicle's gait. Spatial and temporal filtering, along with stereo model based outlier rejection, is used to make the terrain maps accurate and robust. The system works in a wide variety of environments and conditions, including rain and snow, as well as in dynamic environments. A near-infrared illuminator system is used to enable night-time operation. The system explicitly detects sparse vegetation and negative obstacles to enable the full mobility capability of the vehicle.

As LS3 testing continues, the mapping system will be tested on increasingly difficult terrain and conditions. We expect to add the ability to handle standing water and puddles, which can create reflections that appear as negative obstacles. We also plan to continue developing the automatic gait, posture, and footstep adjustments to make the system more robust. To make the system more effective during night operations, we intend to automatically adjust the illuminators to adapt to the terrain geometry. We are also investigating the use of multiple image exposures to increase the dynamic range of the system.

ACKNOWLEDGMENT

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology with funding from the DARPA LS3 program via a subcontract from Boston Dynamics, through an agreement with NASA. The LS3 sensor head was manufactured by CMU's NREC.

REFERENCES

[1] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," vol. 22, no. 6, June 1989.

[2] K. Konolige, "Improved occupancy grids for map building," vol. 4, pp. 351–367, 1997.

[3] C. Urmson and et al., "Autonomous driving in urban environments: Boss and the urban challenge," vol. 25, no. 8, 2008.

[4] J. A. Bagnell, D. Bradley, D. Silver, B. Sofman, and A. Stentz, "Learning for autonomous navigation: Advances in machine learning for rough terrain mobility," vol. 2, no. 2, pp. 74–84, July 2010.

[5] S. Thrun and et al., "Stanley, the robot that won the darpa grand challenge," vol. 23, no. 9, pp. 661–692, July 2006.

[6] E. Marder-Eppstein, E. Berger, T. Foote, B. P. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010.

[7] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: a probabilistic, flexible, and compact 3d map representation for robotic systems," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA) Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010.

[8] J. Lalonde, N. Vandapel, and M. Hebert, "Natural terrain classification using three-dimensional ladar data for ground robot mobility," vol. 23, no. 10, pp. 839–861, October 2006.

[9] C. Plagemann, S. Mischke, S. Prentice, K. Kersting, N. Roy, and W. Burgard, "A bayesian regression approach to terrain mapping and an application to legged robot locomotion," vol. 26, p. 789811, 2009.

[10] J. Chestnutt, Y. Takaoka, K. Suga, K. Nishiwaki, J. Kuffner, and S. Kagami, "Biped navigation in rough environments using on-board sensing," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[11] K. Nishiwaki, J. Chestnutt, and S. Kagami, "Autonomous navigation of a humanoid robot on unknown rough terrain," in *Int. Symposium on Robotics Research (ISRR)*, 2011.

[12] J. Gutmann, M. Fukuchi, and M. Fujita, "3d perception and environment map generation for humanoid robot navigation," vol. 27, no. 10, 2008.

[13] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[14] T. Whelan, J. McDonald, M. Kaess, M. Fallon, H. Johannsson, and J. J. Leonard, "Kintinuous: Spatially extended kinectfusion," in *Robotics: Science and Systems (RSS) RGB-D Workshop*, 2012.

[15] C. Haene, C. Zach, J. Lim, A. Ranganathan, and M. Pollefeys, "Stereo depth map fusion for robot navigation," in *International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[16] M. Bajracharya, A. Howard, L. Matthies, B. Tang, and M. Turmon, "Autonomous off-road navigation with end-to-end learning for the lagr program," vol. 26, no. 1, 2009.

[17] H. Ghazouani, M. Tagina, and R. Zapata, "Robot navigation map building using stereo vision based 3d occupancy grid," vol. 1, no. 3, 2010.

[18] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[19] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: using kinect-style depth cameras for dense 3d modeling of indoor environments," 2012.

[20] L. Matthies and A. Rankin, "Negative obstacle detection by thermal signature," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2003.

[21] R. D. Morton and E. Olson, "Positive and negative obstacle detection using the hld classifier," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011.

[22] N. Heckman, J. Lalonde, N. Vandapel, and M. Hebert, "Potential negative obstacle detection by occlusion labeling," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2007.

[23] "Vegetation detection for driving in complex environments," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2007.

[24] D. V. Nguyen, L. Kuhnert, T. Jiang, S. Thamke, and K. Kuhnert, "Vegetation detection for outdoor automobile guidance," in *IEEE Int. Conf. on Industrial Technology (ICIT)*, 2011.

[25] S. Goldberg, M. Maimone, and L. Matthies, "Stereo vision and rover navigation software for planetary exploration," in *IEEE Aerospace Conference*, 2002.

[26] J. Ma, S. Susca, M. Bajracharya, M. Malchano, D. Wooden, and L. H. Matthies, "Robust multi-sensor, day/night 6-dof pose estimation for a dynamic legged vehicle in gps-denied environments," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[27] L. Matthies and S. Shafer, "Error modeling in stereo navigation," vol. RA-3, no. 3, pp. 239–250, 1987.