A compact, lightweight and energy efficient system for autonomous navigation based on 3D vision

Stefano Mattoccia, Paolo Macrí, Giacomo Parmigiani, Giuseppe Rizza
University of Bologna
Department of Computer Science and Engineering (DISI)
Viale Risorgimento, 2 40136 Bologna, Italy
Email: stefano.mattoccia@unibo.it

***Abstract*—This paper proposes a compact and lightweight system for 3D vision based autonomous navigation. Our navigation system consists of two main components: a stereo vision camera with FPGA processing developed by our research group and an embedded quad core ARM board. These two components allow us to implement a robust obstacle detection algorithm that enables control of a moving vehicle in an unknown (indoor or outdoor) environment. Compared to other vision based navigation system aimed at dealing with this task, our proposal is very compact, lightweight and, thank to the reduced energy requirements, suited for battery powered vehicles. The 3D sensing capabilities provided by the embedded stereo camera deployed make the system very robust. We extensively tested our proposed architecture and navigation system using a small and battery powered rover in very challenging environments. In this paper we provide experimental results concerned with the vision module on sequences acquired in real indoor and outdoor application scenarios. The proposed system is an ideal platform for future developments aimed at 3D registration, visual odometry and object categorization/recognition.**

I. Introduction

Systems for autonomous navigation aim to guide vehicles without human intervention by sensing the environment with appropriate devices. This research field is very challenging and in recent years we have witnessed several progress in this area, in particular concerning *driver-less cars*. In this paper we propose a compact and lightweight computing architecture that enables robust 3D vision based obstacle detection even on small and battery powered autonomous vehicles. The proposed system is based on two main components: a stereo vision [1] camera with FPGA onboard processing [2], [3] and an embedded ARM computing architecture. The power consumption of the whole system (3D camera plus embedded ARM board) is below 10 W and it weights about 130 g (with lenses). The stereo camera provides dense depth maps in real-time according to state of the art stereo vision algorithms mapped on a compact (the processing unit is about 9×4.3 cm) and lightweight (less than 100 g) FPGA board. The quad core ARM embedded board, relies on the depth data provided by the stereo camera to perform robust obstacle detection in the area perceived by the 3D vision sensor. The robust obstacle detection approach proposed relies on two modules working in the disparity domain: a *roll detection* algorithm, aimed at inferring the horizontal tilt angle of the camera with respect to the ground plane, and a *plane segmentation* algorithm, aimed at detecting points belonging to the ground plane. This strategy allows us to control an autonomous vehicle that, for our experimental results, consists in a small battery powered rover with four independent wheels and an Arduino board for motor control connected (via UART) to the ARM board. The overall autonomous navigation system proposed in this paper has been successfully tested on challenging indoor and outdoor environments and provides an excellent platform for further developments in this area.

II. Related work

Different approaches have been proposed in literature for autonomous navigation and they can be divided into two main broad classes; *map based*, using a map of the explored environment (already known or built *on line*) and *map-less*, also referred to as *reactive systems*, which do not need any pre-knowledge of the explored environment. Both approaches rely on information provided by devices that sense the exploration area and, for vision based approaches, information comes from image analysis algorithms (e.g. optical flow, segmentation, etc) or depth maps. Our proposal falls in the map-less class and uses as sensing device the dense disparity field provided by a stereo camera in real-time.

An early solution to the obstacle detection problem was proposed by Thorpe et al [4] using monocular vision to follow the road and 3D vision to avoid obstacles. A mixed 2D and 3D approach was described in [5]; in this case the system relies on single camera and a laser range scanner to sense the environment. In [6] is reported a 2D system aimed at detecting obstacles (vehicles) according to specific patterns found in the images and a simple stereo algorithm for validation. Lorigo et al [7] proposed a 2D low resolution vision system to avoid obstacles making assumptions on the sensed environment. Object detection relies on brightness gradients, RGB color and HSV (hue, saturation, value) information. A technique that uses 3D data was described in [8]; it is based on the calculation of an instantaneous obstacle map which consists of a local occupancy grid that discriminates the presence of obstacles in the sensed area. Kormann et al [9] proposed a method based on assumptions derived from 3D information; they used as input the disparity map and modeled vehicles as cuboids. Labayrade [10] proposed a robust system for floor detection based on an histogram approach (referred to as U-V-disparity) based on 3D data. A similar approach is reported in [11]. The roll detection problem, addressed in this paper by means of a disparity based approach, was previosly tackled [12]

using monocular images and histogram of oriented gradients. Concerning algorithms for robust regression, required by U-V-histogram based approaches for plane segmentation, RANSAC [13] and Hough transform (e.g., in [10]) play a major role in this area. In [14] an approach aimed at improving effectiveness of the cost function used in the original RANSAC approach was proposed. Finally, [15] described a verification test for the RANSAC framework aimed to speed-up convergence. A performance evaluation of different RANSAC approaches applied to planar homography estimation can be found in [16].

III. Proposed computing architecture

Our vision based autonomous navigation system is mainly aimed to application scenarios characterized by constrained energy requirements. In particular, we aim to address the requirements of small battery powered vehicles. Therefore, major constraints in our work are power consumption, weight and size. Our system basically consists of two devices: a compact stereo camera with FPGA onboard processing and a quad core ARM embedded system.

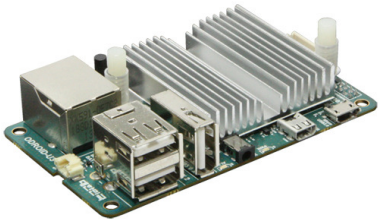

Fig. 1. The quad core ARM embedded system Odroid U3 [17] used in our navigation system to process, at about 20 fps, the 3D data provided by the stereo camera.

A. Embedded 3D camera

The stereo camera, developed by our research group [2], [3] , processes a synchronized video stream acquired by two global shutter wide VGA (i.e. 752×480 resolution) imaging sensors at more than 30 fps. This device allows us to obtain dense depth map according to state of the art stereo vision algorithms. The overall stereo vision pipeline (including *rectification*, *stereo matching*, *subpixel disparity estimation* and *filtering*) is mapped into a low cost Xilinx Spartan 6 FPGA. For our experiments, we configured the stereo camera with a baseline (distance between the two optical sensors) of 6 cm, monochrome sensors, 320×240 image resolution and mapped into the FPGA our custom implementation of the SGM algorithm [18]. The disparity maps provided by the stereo camera are aligned to the rectified reference frames. The stereo camera is connected to the host computer (i.e. the ARM board in our setup) by means of a standard USB 2.0 interface. The camera requires about 2 W, and for this reason it is self-powered through the same USB data cable. The processing module containing the FPGA has an area smaller than a credit card and the weight of the whole stereo camera, including imaging sensors and lenses, is less than 90 g making this device well suited to embedded applications.

B. Embedded processing

The other essential device in our vision system module is an embedded board (specifically, for our experiments, the ODROID-U3 [17] shown in Figure 1); this device contains an ARM SoC, model Exynos 4412, Cortex-A9 architecture with four physical cores at 1.7 GHz, 1 MB level 2 cache, 2GB of LP-DDR2 880 MHz RAM, a USB controller and an Ethernet controller. The embedded platform shown in Figure 1) has a size of 83 x 48 mm and weights less than 50 g including the heat sink. The operating system installed is Linux Xubuntu 13.10. The energy requirement of this platform is below 10 W (typically about 5W), making this device an ideal candidate for a navigation system.

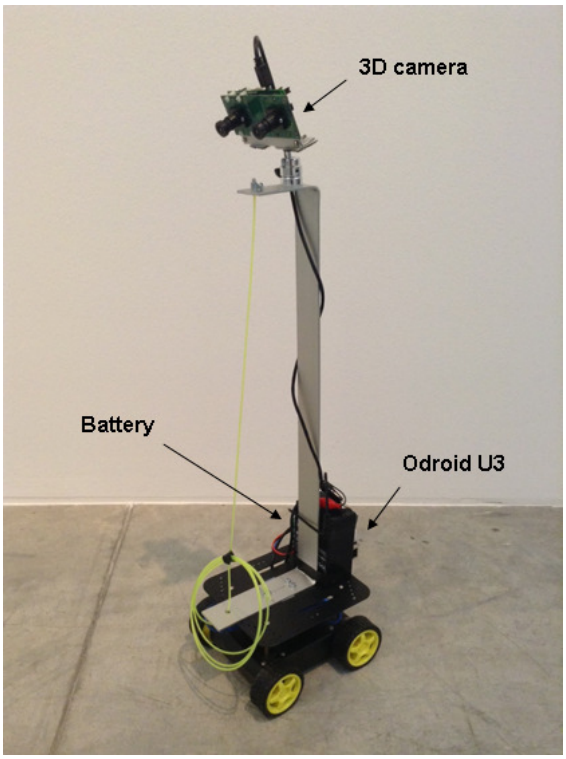

Fig. 2. Overall system used for experimental results. The FPGA-based stereo camera is located at the top of the small battery powered rover.

Although not an essential part of the navigation system proposed, the vehicle used for the experimental validation is based on a simple rover with four independent wheels and motors, an Arduino board that controls the motors according to commands sent via an UART interface by the vision controller (i.e. the stereo camera plus the ARM embedded board) and batteries. The overall system used for our experimental results is shown in Figure 2. At the top of the figure we can notice the 3D camera and, at the bottom, the large battery used for preliminary experiments. The Odroid, not visible in Figure 2 is placed behind the battery.

IV. Robust obstacle detection

For our obstacle detection strategy, is fundamental the identification of the ground plane in the area sensed by the 3D camera. This task is often referred to as *plane segmentation*, although ground plane detection consists in a specific case of the more general plane segmentation problem. Once completed this task, the autonomous vehicle can easily identify approaching obstacles as those points not belonging to the ground plane. Of course, robust obstacle detection methodologies capable to

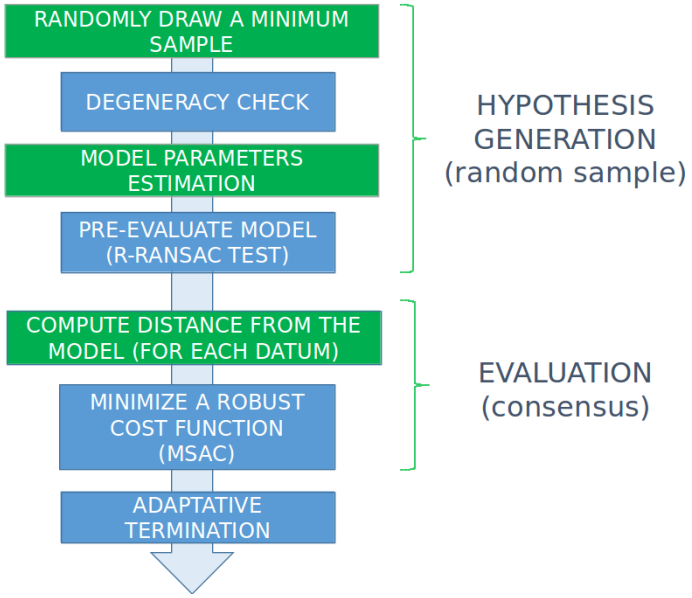

Fig. 3. Main modules of our RANSAC framework. In green the standard modules of the RANSAC technique and in blue the additional modules included in our framework.

deal with potentially noisy data are mandatory in this case. For this purpose, our approach relies on the disparity maps provided by the stereo camera and appropriately processed by means of a robust plane detection algorithm. It is worth observing that, in the depicted application scenario, in most cases the vehicle will move on a flat ground plane and hence the ground plane could be detected only once at startup. However, in a more generic scenario and especially in outdoor environments, the navigable area on which the vehicle moves is often characterized by slightly slanted surfaces. Therefore, our control unit detects within the sensed area the points belonging to the ground plane in each frame acquired by the 3D vision sensor.

Another problem arises when the horizontal axis of the 3D sensor is not aligned with the ground plane. This problem might occur due to an inaccurate initial setup of the 3D sensor with respect to the ground plane. Other reasons for this misalignment might occur in presence of small obstacle/hole under the wheels of the vehicle or when the 3D sensor changes its original position (pan/tilt/rotation) by means of appropriate motor units. Regardless to the reasons of this potential misalignment, we aim to solve this problem by means of a vision based *roll detection* algorithm, using the information provided by the 3D vision sensor.

A. Ground plane detection

Among the many strategies proposed in literature to robustly solve the plane segmentation task, we use an efficient and robust method that processes disparity information in a sub-dimensional space. Specifically, we adopt an histogram based approach that processes in real-time the dense disparity maps provided by the 3D camera. This method, originally proposed in [10] and referred to as V-disparity, is quite common for obstacle detection in automotive applications. A viable alternative to this approach would consist in a robust fitting of 3D points obtained by mapping the disparity information into a point cloud according to the calibration parameters of the stereo camera. Working in the point cloud domain would certainly be effective; nontheless, compared to our histogram based approach, more demanding in terms of computational requirements and memory footprint.

Given a 3D sensor with its horizontal axis aligned or almost aligned with the ground plane, the v-disparity approach relies on the following observation: in the disparity map, points belonging to the ground plane are projected into a set that can be modeled by a line segment in the histogram domain. Figure 4 shows, on the left, the rectified reference image of a scene containing some objects laying on an almost uniform ground plane and, in the middle, the corresponding disparity map provided by the stereo camera. In the right portion of the figure we can see the resulting V-disparity histogram; we can easily observe that the highlighted line corresponds to the ground plane in the disparity map. In the upper area of the V-disparity histogram we can also notice the line corresponding to the ground plane being perturbed by other objects in the scene such as the shopping bag, the trash-can and the furnitures in the background. Furthermore, since disparity maps provided by a 3D sensor are generally noisy, a robust regression technique for ground detection able to deal with all these problems is essential. This task can be accomplished by means of different robust regression techniques; however, considering the embedded application scenario outlined, RANSAC [13] based approaches appear optimal candidates.

For the reasons outlined so far, we implemented a custom RANSAC framework made up of different pluggable modules in order to assess the effectiveness of recent improvements proposed in this field. Figure 3 shows the main modules of our regression framework, displaying in blue the add-ons with respect to the original RANSAC algorithm (in green). Our framework, compared to the original RANSAC approach, can handle a larger amount of outliers. In particular, it contains the following additional features:

- A degeneracy check loop inside sampling phase, to avoid degenerate model hypothesis, like collinear vertical points in case of a straight line
- An R-Ransac [15] Td,d test switch that can be optionally triggered in the sampling stage, enabling a preverification stage for fast rejection of bad hypotheses
- A more effective cost function for the RANSAC minimization problem as proposed in [14]
- An optional adaptive termination criteria that enables to determine the maximum number of iterations on every cycle [16]

Each additional feature available in our RANSAC framework can be enabled/disabled in order to assess its effectiveness.

B. Roll detection

The roll detection algorithm aims at inferring the rotation angle between the horizontal axis of the 3D camera and the ground plane. Although the cost of MEMS (Micro Electro-Mechanical Systems) based devices, such as accelerometers

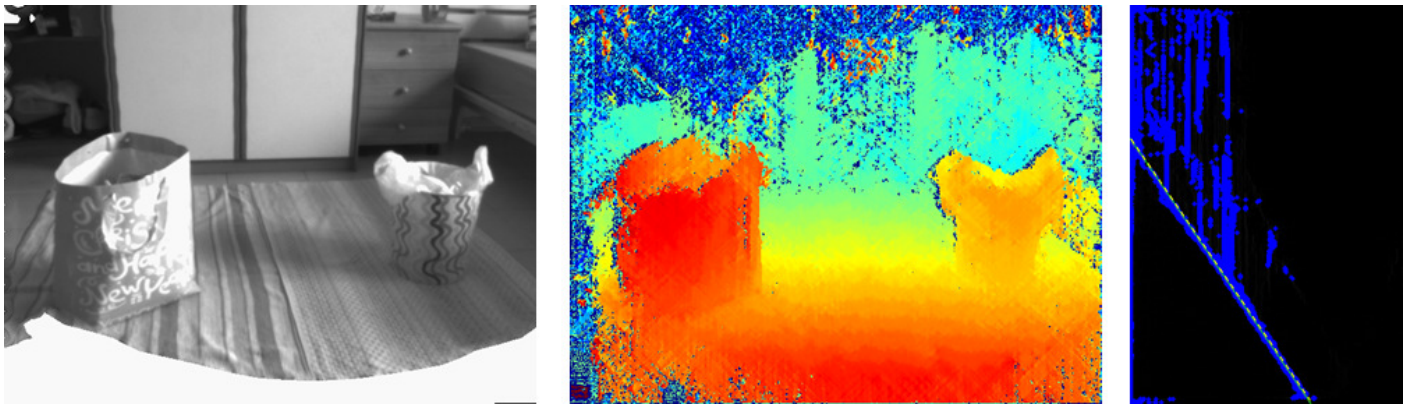

Fig. 4. Left, rectified reference image - Center, disparity map (warmer colors correspond to points closer to the stereo camera) - Right, V-disparity with highlighted the line corresponding to ground points

and gyroscopes, has dropped in recent years, some authors (e.g. [12]) pointed out that an Inertial Measurement Unit (IMU) might be in some circumstances *unreasonably expensive*. For this reason, although our stereo camera could be equipped with an IMU using an expansion module, for our preliminary experimental results we preferred a vision based roll-detection algorithm. Eventually, this strategy would be feasible for data fusion by integrating an IMU.

Our vision based roll detection algorithm works in the disparity space and, similarly to the V-disparity histogram approach [10], assumes that the sensed disparity map contains a sufficiently extended portion of the ground plane. Moreover, and similarly to the plane segmentation algorithm, our roll detection approach relies on the same robust RANSAC framework described in the previous section. In the first step we select a region of interest (ROI), where we expect that ground points are more likely to be, and determine the most frequent disparity value (i.e. the *mode*) within this area. Then, we select within the ROI a subset of disparity values close (according to a defined threshold) to the mode and robustly determine by means of our RANSAC framework applied to these points, potentially contaminated by gross errors (i.e. outliers, points not belonging to the road surface), the line that best fits with the *filtered* measurements. Finally, from the slope of the fitted line we can then easily retrieve the roll-correction angle and thus perform a rotation of the original disparity image.

As ROI we selected for our experiments an area centered in the middle of the disparity image, assuming that in this region we are more likely to find a sufficiently extended area belonging to the ground plane. Clearly, setting as ROI an area very close to robot (and thus, not containing objects thank to the obstacle navigation system) would probably avoid at all objects within the selected ROI. Despite this fact, the proposed approach could be further improved by selecting multiples ROIs and/or by using a voting scheme. In order to further improve the effectiveness of the roll detection algorithm described so far, a framework based on Kalman filter could be used to reduce the effect of sporadic failures. Moreover, the same Kalman framework could be eventually used to integrate the information provided by the vision based roll-detection algorithm and the IMU.

The roll detection algorithm described can produce substantial benefit to the overall obstacle detection approach in the presence of a non-negligible rotation of the camera with respect to the ground plane. Compared to monocular approaches, such as those using gradient orientation histograms [12], roll detection based on disparity values is less prone to artifacts that can arise in the 2D image domain and hence potentially more robust.

V. Obstacle detection and control

In order to enable a vehicle to move autonomously in an unknown environment is mandatory to detect obstacles and provide appropriate commands as fast as possible to the motor unit. In our current control strategy, starting from the robust plane segmentation approach described in the previous section, we identify obstacles as cluster of objects not belonging to the ground plane. If the detected obstacles fall into a specific sensed area of the scene corresponding to the lateral extent of the rover, the vision based control unit sends to the vehicle (via an UART interface) a command to the motors (managed, in our current rover, by a compact Arduino micro controller) in order to avoid the detected obstacles. Thank to the onboard FPGA processing, the embedded system (the Odroid U3 in our experiments) acquires and processes disparity maps at about 20 fps with a minimal overhead including image visualization. This allows the ARM embedded system, without deep code optimizations, to detect obstacles as well as to handle motor control (through the Arduino module) at a similar frame rate. The overall computation can be carried out by using just a fraction of the computational resources available in the quad core embedded processor.

VI. Experimental results

In this section we report experimental results aimed at assessing the performance of the vision module proposed. To this aim we focus our attention on the robust obstacle and roll detection methodology outlined in the previous sections. The overall system, made of the vision based module (3D camera plus ARM embedded system) and the battery powered vehicle described in the previous sections, was extensively tested on the field, including a successful two days demonstration in an indoor environment with many people moving in the same area

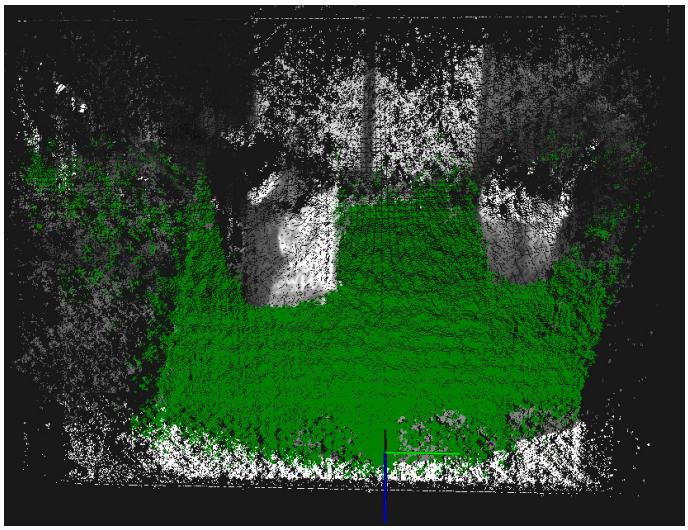

Fig. 5. Plane segmentation obtained processing the pointcloud corresponding to the frame of the dataset DS1 shown in Figure 4. The ground plane is highlighted in green.

of the vehicle[1]

A. Plane segmentation: disparity histogram vs pointcloud

In this experiment we compare, on a dataset, referred to as DS1 (one frame is shown in Figure 4) and made of several frames acquired in an indoor scenario with our 3D camera, the disparity histogram approach proposed in this paper with the PCL [19] robust regression module based on point cloud data.

For this latter approach we extracted the point cloud using the calibration parameters of the stereo camera and on this data structure applied the optimized plane segmentation function provided by PCL. The outcome of the PCL approach, for one frame of dataset DS1 (the same frame depicted in Figure 4), is shown in Figure 5 with the detected ground plane highlighted in green). For both approaches we set the maximum number of iterations k for the RANSAC algorithm to 100 and for each frame of the dataset we averaged the execution time for 1000 iterations. The testing machine was a Linux platform and for our histogram based approach we didn't apply any specific code optimizations (e.g. SIMD instructions or multi core processing capabilities available in our quad core processing platform). Compared to the disparity histogram algorithm, the point cloud approach does not require a roll detection algorithm being able, by robustly fitting by means of RANSAC based approaches the raw 3D data (i.e. the point cloud), to detect slanted planes without any problem. Regardless of this problem, handled in our current approach by means of a specific roll-detection algorithm, our solution, being three times faster, clearly outperforms the PCL counterpart in terms of execution time. The memory footprint required to store the pointcloud is also significantly higher compared to our histogram based approach. Concerning the effectiveness, both approaches always correctly find the ground planes in the considered dataset. However, since computational requirements represent a major issue for embedded platforms, the histogram based approach outlined in this paper clearly represents the optimal solution.

[1]A video concerned with this evaluation is available at this link: https://www.youtube.com/watch?v=7rieq3wfGDo

B. Comparison of robust regression methods

In this section we compare the robust regression approach described in this paper with other approaches, within the same histogram based methodology for plane segmentation, on two different datasets: DS1, the same indoor dataset used for previous experimental results (one frame is shown in Figure 4), and dataset DS2 [20], available on line [21], concerned with an outdoor environment.

More explicitly, we experimentally tested on the two datasets different improvements to the basic RANSAC approach [13] according to the framework depicted in Figure 3. Our evaluation included the R-RANSAC approach [15] determining, however, that the M-SAC approach proposed by Torr and Zisserman [14] grants the trade-off between robustness (it enables to correctly determine the ground plane on each frame) and computational requirements (resulting the faster method on the chosen testing platform, for both datasets). With the M-SAC approach, the adoption of an adaptive termination criteria [16] maintains the same effectiveness but leads to a variable execution time. For this reason we preferred a solution based on a maximum (k=100 in our experiments) number of iterations that has a fixed upper bound on the execution time. We also compared the outlined RANSAC regression approach based on the M-SAC variant for plane segmentation with the *cvFitLine* function available in the OpenCV library. This function relies on the Iteratively Re-weighted Least Square (IRLS) approach with an M-estimator as cost function and has a fixed number of maximum iterations. For our approach we set again a number of iterations k=100 and measured the execution time, averaged on 500 iterations on the two datasets DS1 and DS2. In both datasets our approach correctly enables to determine the ground plane on each frame. On the other hand, the M-estimator approach adopted by the cvFitLine function, on dataset DS2, containing a huge presence of outliers, is unable to determine the ground plane almost on each frame of the sequence. Differently, on dataset DS1 the cvFitLine function correctly determines in each frame the ground plane. Examining the execution times, our proposal is on average 30% slower than the cvFitLine function, being however much more reliable considering both datasets. We tested our RANSAC based approach for regression with a method based on the Hough transform determining that, although both methods have the same effectiveness, the latter method is more computationally demanding. Finally, on our target platform and processing disparity maps provided by the 3D camera configured at 320×240 resolution, the overall execution time, using only one CPU core, is 13.4 ms enabling real-time obstacle detection on the Odroid U3.

C. Roll detection

Finally, we report experimental results regarding the proposed roll detection algorithm, showing here the output for one frame of the DS1 dataset. In this case, the disparity map is encoded in greyscale; brighter color correspond to points closer to the 3D sensor. Figure 6 shows a synthetically rotated disparity map and the output of the roll-detection algorithm followed by the plane segmentation approach. Observing the

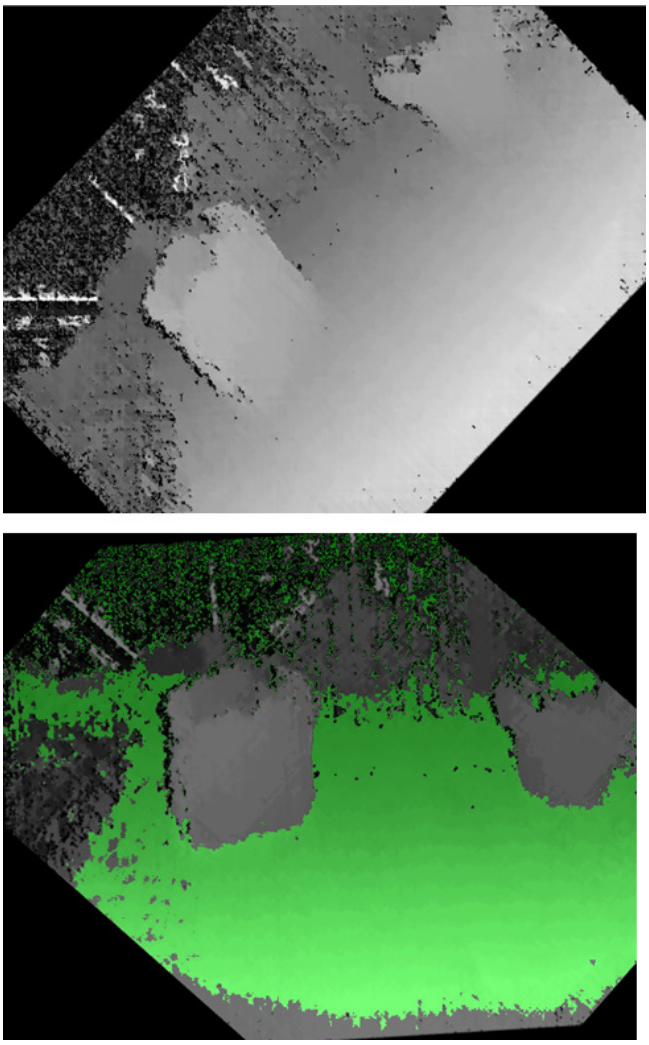

Fig. 6. Top, synthetically rotated disparity map - Bottom, outcome of roll detection algorithm and plane segmentation (highlighted in green).

figure we can notice that the roll-detection algorithm can effectively handle a quite large rotation of the horizontal axis of the 3D camera with respect to the ground plane.

The roll detection approach was successfully tested also on DS2 with similar synthetic rotation imposed to the input disparity. Nevertheless, as previously pointed out, adding a Kalman filtering stage would further increase the reliability of the roll detection algorithm. On the Odroid platform the roll detection algorithm takes 6.7 ms.

VII. Conclusion

In this paper we have described a compact and lightweight vision system for autonomous vehicle, suited for embedded robotic applications, based on a stereo camera with onboard FPGA processing. Our current vision system tackles the problem of robust obstacle detection for real-time autonomous navigation. The overall vision system weights less than 130 g and, compared to similar systems based on more standard computing architectures, has a reduced power consumption (i.e. less than 10 W considering the quad core embedded ARM platform and the stereo camera). This enables its deployment in applications characterized by constrained power requirements such as those involving battery powered small robots or drones. Future work is aimed at adding to the vision system proposed SLAM (Simultaneous Localization and Mapping) capabilities in order to perform visual odometry as well as 3D reconstruction of the scene sensed by the 3D camera. Finally, we also aim at adding semantic perception capabilities to the vision system. In particular we are currently interested in enabling object categorization in order to recognize specific class of objects while the vehicle is moving in unknown environments.

References

[1] R. Szeliski, *Computer Vision: Algorithms and Applications*, 1st ed. New York, NY, USA: Springer-Verlag New York, Inc., 2010.

[2] S. Mattoccia, I. Marchio, and M. Casadio, "A compact 3d camera suited for mobile and embedded vision applications," in *4th Mobile Vision Workshop*, Columbus, OH (USA), 2014.

[3] S. Mattoccia, "Stereo vision algorithms for fpgas," in *9th Embedded Vision Workshop*, Portland, OR (USA), 2013.

[4] C. Thorpe, M. H. Hebert, T. Kanade, and S. A. Shafer, "Vision and navigation for the carnegie-mellon navlab," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 3, pp. 362–372, May 1988.

[5] M. Turk, D. G. Morgenthaler, K. D. Gremban, and M. Marra, "Vits-a vision system for autonomous land vehicle navigation." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 3, pp. 342–361, 1988.

[6] M. Bertozzi, A. Broggi, A. Fascioli, and S. Nichele, "Stereo Vision-based Vehicle Detection," in *Procs. IEEE Intelligent Vehicles Symposium 2000*, Detroit, USA, Oct. 2000, pp. 39–44.

[7] L. M. Lorigo, R. A. Brooks, and W. E. L. Grimson, "Visually-guided obstacle avoidance in unstructured environments," in *IEEE Conference on Intelligent Robots and Systems*, 1997, pp. 373–379.

[8] S. Badal, S. Ravela, B. A. Draper, and A. R. Hanson, "A practical obstacle detection and avoidance system." in *WACV*. IEEE, 1994, pp. 97–104.

[9] B. Kormann, A. Neve, G. Klinker, and W. Stechele, "Stereo vision based vehicle detection." in *VISAPP (2)*, 2010, pp. 431–438.

[10] R. Labayrade and D. Aubert, "In-vehicle obstacles detection and characterization by stereovision," in *in Proceedings the 1st International Workshop on In-Vehicle Cognitive Computer Vision Systems*, 2003, pp. 13–19.

[11] Y. Gao, X. Ai, Y. Wang, J. Rarity, and N. Dahnoun, "U-v-disparity based obstacle detection with 3d camera and steerable filter." in *Intelligent Vehicles Symposium*, 2011, pp. 957–962.

[12] M. Schlipsing, J. Schepanek, and J. Salmen, "Video-based roll angle estimation for two-wheeled vehicles," in *Intelligent Vehicles Symposium*, 2011, pp. 876–881.

[13] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[14] P. H. S. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, p. 2000, 2000.

[15] J. Matas and O. Chum, "Randomized ransac with td, d test." *Image Vision Comput.*, vol. 22, no. 10, pp. 837–842, 2004.

[16] S. Choi, T. Kim, and W. Yu, "Performance evaluation of ransac family," in *BMVC*, 2009.

[17] Hardkernel, http://hardkernel.com/main/main.php.

[18] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, 2008.

[19] http://pointclouds.org/.

[20] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme," in *Intelligent Vehicles Symposium (IV)*, 2010.

[21] https://www.youtube.com/watch?v=sJGYzJbf6iI.