Collapsible Cubes: Removing Overhangs from 3D Point Clouds to Build Local Navigable Elevation Maps

Antonio J. Reina, Jorge L. Martínez, Anthony Mandow, Jesús Morales, and Alfonso García-Cerezo

Departamento de Ingeniería de Sistemas y Automática
University of Málaga, 29071 Málaga, Spain. Email: ajreina@uma.es

Abstract - Elevation maps offer a compact 2 ½ dimensional model of terrain surface for navigation in field mobile robotics. However, building these maps from 3D raw point clouds con- taining overhangs, such as tree canopy or tunnels, can produce useless results. This paper proposes a simple processing of a ground-based point cloud that identifies and removes overhang points that do not constitute an obstacle for navigation while keeping vertical structures such as walls or tree trunks. The procedure uses efficient data structures to collapse unsupported 3D cubes down to the ground. This method has been successfully applied to 3D laser scans taken from a mobile robot in outdoor environments in order to build local elevation maps for navigation. Computation times show an improvement with respect to a previous point-based solution to this problem.

Keywords: Mobile robots, terrain modelling, field robotics, 3D point clouds.

Citation Information

Antonio J. Reina, Jorge L. Martínez, Anthony Mandow, Jesús Morales, and Alfonso García-Cerezo,
" Collapsible Cubes: Removing Overhangs from 3D Point Clouds to Build Local Navigable Elevation Maps",
IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp.1012-1017, 2014

Collapsible Cubes: Removing Overhangs from 3D Point Clouds to Build Local Navigable Elevation Maps

Antonio J. Reina, Jorge L. Martínez, Anthony Mandow, Jesús Morales, and Alfonso García-Cerezo
Universidad de Málaga, Andalucía Tech, Dpto. Ingeniería de Sistemas y Automática, 29071-Málaga, Spain
Email: ajreina@uma.es, Tel: (+34) 951 952 251.

***Abstract*— Elevation maps offer a compact $2\frac{1}{2}$ dimensional model of terrain surface for navigation in field mobile robotics. However, building these maps from 3D raw point clouds containing overhangs, such as tree canopy or tunnels, can produce useless results. This paper proposes a simple processing of a ground-based point cloud that identifies and removes overhang points that do not constitute an obstacle for navigation while keeping vertical structures such as walls or tree trunks. The procedure uses efficient data structures to collapse unsupported 3D cubes down to the ground. This method has been successfully applied to 3D laser scans taken from a mobile robot in outdoor environments in order to build local elevation maps for navigation. Computation times show an improvement with respect to a previous point-based solution to this problem.**

I. Introduction

Mapping unknown environments from three-dimensional (3D) point clouds is a key issue for autonomous vehicle navigation in outdoor applications [1]. A popular method to represent terrain is to build elevation maps [2], which provide a computationally effective navigation model of the environment. These compact $2\frac{1}{2}$ maps have been represented as regular grids [3] [4] or as irregular triangular meshes [5] [6]. However, using raw point clouds with overhangs to build elevation maps can produce unreliable terrain models.

Some overhangs, like tree canopy or passages under bridges, should not be considered an obstacle for navigation. Thus, identification of traversable overhangs has been addressed with extended elevation maps by computing the variance of the height for all the points that fall on each cell of a two-dimensional (2D) grid [3] [7]: If variance exceeds a given threshold, cell points are processed to search for vertical gaps. With a higher computational cost, multi-level maps identify overhangs to account for multiple elevations [8] [9]. In [5], the slope of triangles in a mesh is checked to preserve only those that are nearly-flat, which removes walls and ceilings. On the other hand, overhangs can be considered as obstacles when the vertical gap is not enough for the robot to pass through [10]. Furthermore, considering height differences as obstacles is reasonable for some natural terrains, like desert or planetary environments, where overhangs are not present [11].

Identifying and removing overhangs from raw point clouds can be regarded as a classification problem, where it is necessary to distinguish at least between ground and overhangs [12]. Further processing can classify some obstacles as vegetation [13] [14] [15]. Classification involves a large amount of sparse 3D data from complex environments obtained from 3D scanners [3] [7] [12] or stereo vision [16] [17]. Therefore, efficient data structures and algorithms are required for 3D data processing [18]. In a previous work, we proposed efficient data structures for fast and precise pairwise point-cloud registration based on 3D coarse binary cubes (CBCs) [19].

This paper proposes a new use of CBC data structures to implement a simple principle for identification and removal of overhangs from a ground-based point cloud. Instead of a point-based gap search [3] [7], the principle is based on collapsing unsupported 3D cubes down to the ground. The proposed cube-based method has been tested with 3D laser scans taken from a mobile robot in outdoor environments, where the resulting point clouds have been used to compute both standard and fuzzy elevation maps. Furthermore, the computation time of the new method is compared with the point-based strategy.

The rest of the paper is organized as follows. Next section reviews CBC data structures. The proposed procedure to remove overhangs is described in section III. Experimental results are presented in section IV. Finally, last section is devoted to conclusions and future work.

II. CBC data structures

This section reviews efficient CBC data structures [19], that will be employed in the proposed implementation of the collapsible cubes principle.

To represent a 3D spatial grid composed of cubes of the same edge length E, whose Boolean values are set to one if any Cartesian points of the scan fall within their limits, CBC employs the following data structures:

- V is a binary vector whose elements correspond to the cubes contained by the axis-aligned minimum bounding box for the point cloud. Each cube has a unique integer index I in V.
- L is an unsorted integer list that contains the same information as V in a non-sparse way. The length ℓ of L corresponds to the number of cubes set to one, which is always less or equal than the number n of points from the scan. Moreover, the relation $\ell \ll n$ holds with coarse cubes.

It is assumed that the frame of the 3D point cloud has its Z axis pointing upwards. Let x_{max}, x_{min}, y_{max}, y_{min}, z_{max}, z_{min} be the Cartesian coordinates of the minimum bounding box for the point cloud. Then, each leveled Cartesian point (x, y, z) is indexed with the I value of its corresponding cube:

$$I = i_x + i_y\, i_{xmax} + i_z\, i_{xmax}\, i_{ymax}, \tag{1}$$

where

$$i_{xmax} = \text{round}\left(\frac{x_{max} - x_{min}}{E}\right) + 1, \tag{2}$$

$$i_{ymax} = \text{round}\left(\frac{y_{max} - y_{min}}{E}\right) + 1, \tag{3}$$

and i_x, i_y, i_z are the integer cube coordinates:

$$i_x = \text{round}\left(\frac{x - x_{min}}{E}\right), \tag{4}$$

$$i_y = \text{round}\left(\frac{y - y_{min}}{E}\right), \tag{5}$$

$$i_z = \text{round}\left(\frac{z - z_{min}}{E}\right). \tag{6}$$

In this way, an indexed Cartesian point can be defined by coordinates (x, y, z, I), where several points can have the same I value, meaning that they belong to the same cube. An I value univocally relates to cube coordinates (i_x, i_y, i_z) as:

$$i_x = \text{remainder}\left(\frac{I}{i_{xmax}}\right), \tag{7}$$

$$i_y = \text{remainder}\left(\frac{(I - i_x)/i_{xmax}}{i_{ymax}}\right), \tag{8}$$

$$i_z = \frac{I - i_x - i_y\, i_{xmax}}{i_{xmax}\, i_{ymax}}. \tag{9}$$

III. Collapsing Cubes Method

A. Principle

The goal is to clasify 3D points as ground (including vertical obstacles) and overhangs. The collapsing cubes principle is illustrated in Fig. 1. The cubes of each grid column are processed from the bottom to the top. The first occupied cube is initially considered as the ground. Then, if the distance between this cube and the next occupied cube is lesser than a certain threshold σ, the upper cube becomes the ground. Otherwise, if the next occupied cube is at a distance greater than σ, it is collapsed down to the ground (i.e., removed) and its points are classified as belonging to an overhang. In field mobile robots, σ should be defined to be greater than the height of the robot.

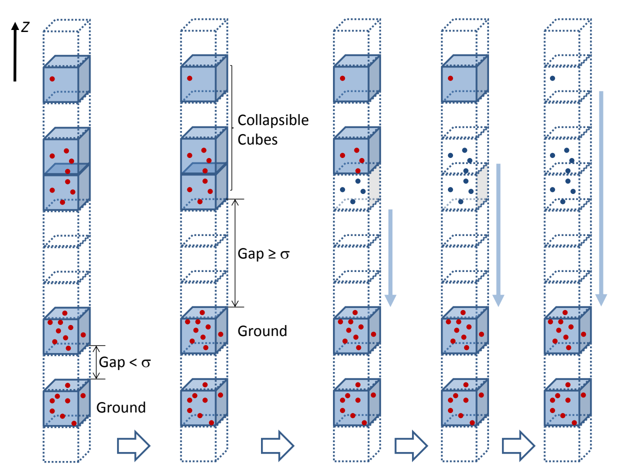

Fig. 1. Steps of the collapsible cubes principle applied to a single column. Points classified as overhangs are depicted in blue color.

B. Procedure

The complete processing of a leveled point cloud is depicted in the flowchart of Fig. 2. CBC data structures are employed as a device to identify and remove overhangs from a point cloud.

Initially, V is a zero vector, and L is empty. A single iteration for all 3D points serves to compute their corresponding I index, as well as to build V and L as follows: If $V(I) = 0$, then $V(I)$ is set to one, and I is inserted into L. Otherwise, V and L remain unmodified. After these data structures are computed, Algorithm 1 is applied to obtain a list of removed cubes R that will be employed to discard overhang points.

Algorithm 1 begins by sorting L in increasing order of I, which, according to (1), implies placing the lowest occupied cubes in the first place. Thus, the repeat loop for all elements in L proceeds upwards as proposed in Section III-A. Moreover, as columns are processed all together, matrix M is defined to keep track of ground cubes at each column. M is initialized with negative values to represent no known ground. This default value is substituted by the index i_z of the first occupied cube in the corresponding column.

For columns that already have a ground index, updating this index depends on σ, which is defined as the minimum number of empty cubes between the ground and overhangs.

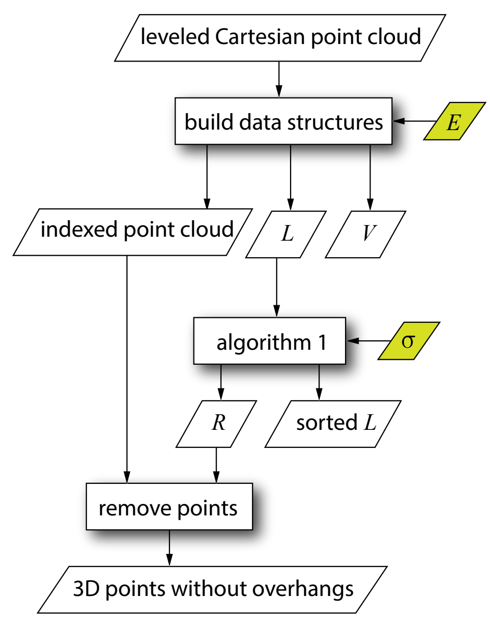

Fig. 2. Flowchart for processing a point cloud to discard overhangs.

Algorithm 1: The collapsible cubes algorithm

Input: L, σ
Variables: M, I, R
Initialize $M \leftarrow -1$ % Matrix of size $i_{xmax}\, i_{ymax}$
$R \leftarrow \{\}$ % List of removed cubes
Sort list L in increasing order of indices I
Access to the first index I of L
repeat
  $(i_x, i_y, i_z) \leftarrow I$ % Using (7)-(9)
  if $M(i_x, i_y) = -1$ **then**
    $M(i_x, i_y) = i_z$
  else if $(i_z - M(i_x, i_y)) \geq \sigma$ **then**
    $R \leftarrow \{R, I\}$ % Let the cube collapse
  else
    $M(i_x, i_y) = i_z$
  Access to the next index I of L
until $I == NULL$;
Result: ***Sorted list*** R ***with cubes from overhangs***

When the gap between an occupied cube and the ground is less than σ, its i_z becomes the new ground. Otherwise, the cube is collapsed by adding its I index to the list of removed cubes R.

All in all, the computational load of the whole process is of order $O(l \log(l))$ that originates from sorting the list L since CBC data structures can be built in $O(n)$ time [19].

IV. Experiments

A. Experimental Setup

This section discusses the implementation of the proposed method. For this purpose, two outdoor terrains have been scanned (see Fig. 3) with a 3D rangefinder built by pitching a Hokuyo UTM-30LX 2D rangefinder. This device has the following specifications: field of view, $270° \times 135°$; maximum range of $30\,\mathrm{m}$; minimum range of $0.1\,\mathrm{m}$; horizontal resolution of $0.25°$; and vertical resolution adjustable from $0.067°$ to $4.24°$ [20]. The scan times at minimum and maximum resolution are $1.54\,\mathrm{s}$ and $95.75\,\mathrm{s}$, respectively. In particular, scans have been obtained with a vertical resolution of $0.278°$, which is similar to the horizontal resolution. The resulting range images have a maximum of $n = 505036$ points, with a scanning time of $12.43\,\mathrm{s}$.

The laser rangefinder is mounted on the 4-wheel skid-steer mobile robot Quadriga [21], $0.7\,\mathrm{m}$ above the ground (see Fig. 4). Other onboard sensors are an IMU and a pan&tilt camera for tele-operation. The robot has a height of $0.9\,\mathrm{m}$.

Fig. 3. Experimental sites: park with trees (first, top) and a tunnel (second, bottom).

Fig. 4. The 3D laser scanner mounted on the Quadriga mobile robot.

B. Classification results

In the experiments, point clouds are obtained in local sensor coordinates, with the Y and Z axes pointing forwards and upwards, respectively. This frame has been transformed according to roll and pitch angles given by Quadriga's IMU for leveling the data. Leveled point clouds for both experiments can be observed in Fig. 5. For the sake of clarity, experiments only use data from a rectangle defined by $x_{min} = -10\,\mathrm{m}$, $x_{max} = 10\,\mathrm{m}$, $y_{min} = 0\,\mathrm{m}$, and $y_{max} = 10\,\mathrm{m}$. Farther ranges have sparse points that would be difficult to observe in the figures. The minimum and maximum vertical point coordinates are $z_{min} = -1.16\,\mathrm{m}$ and $z_{max} = 3.18\,\mathrm{m}$, for the first scan and $z_{min} = -1.12\,\mathrm{m}$ and $z_{max} = 5.35\,\mathrm{m}$ for the second.

The size of the cube edge has been set to $E = 0.5\,\mathrm{m}$. Based on this value, the classification threshold has been chosen as a function of robot height: $\sigma = \mathrm{ceil}(0.9\,\mathrm{m}/E) = 2$, which means that the minimum vertical gap for an overhang is composed of two cubes. Fig. 6 shows in detail cube classification for a tree of the first scene.

Points classified as overhangs are shown with blue color in Fig. 5. In the first scene, these points belong to tree canopy, while in the second case they belong to the ceiling of the tunnel. Note that vertical structures like walls and tree trunks have not been classified as overhangs. Points from some small portions of the tunnel ceiling and tree canopy have

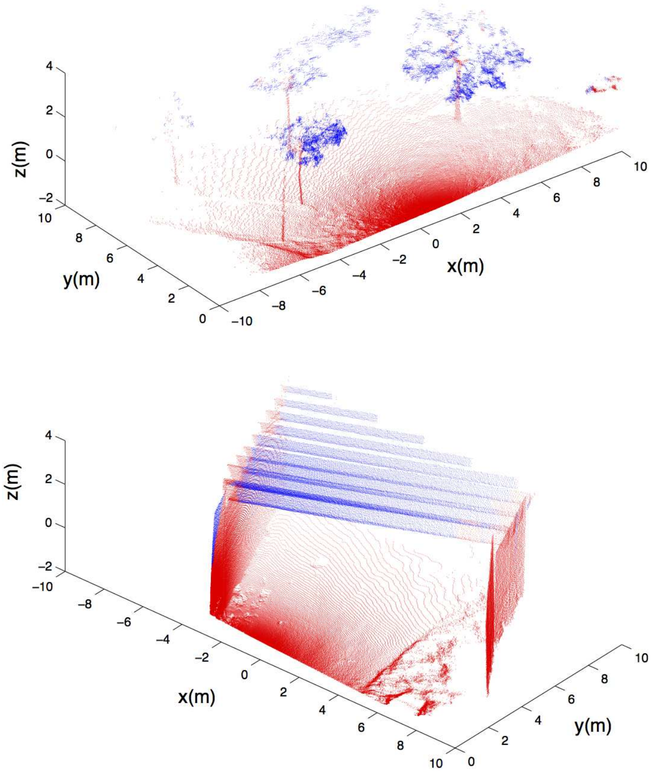

Fig. 5. Leveled point clouds for the first (top) and second (bottom) outdoor scenes. In blue and red colors, discarded and retained points, respectively.

TABLE I
COMPUTATION TIMES.

Method / Scene	First	Second
Cube-based	0.21 s	0.37 s
Point-based	0.39 s	0.54 s

been misclassified as ground (i.e., have not been removed) in regions where ground readings are not available due to sensor shadows or occlusions.

Table I shows computation times for the cube-based and point-based [3] [7] methods. Both methods have been implemented with efficient matrix manipulation functions in Matlab, running in an Intel Core i7 2720QM. The same results are obtained in both cases, but better computation times are achieved by collapsing cubes. The first scene takes less time because of a greater amount of sky measurements that are not processed.

C. Application to Build Elevation Maps

The collapsible cubes method produces a point cloud which keeps all the scan information without the overhang points. This section proposes the application of the point clouds from the two experimental scenes to build digital elevation maps for robot navigation with two different methods. First, standard digital elevation maps (DEMs) can be built by

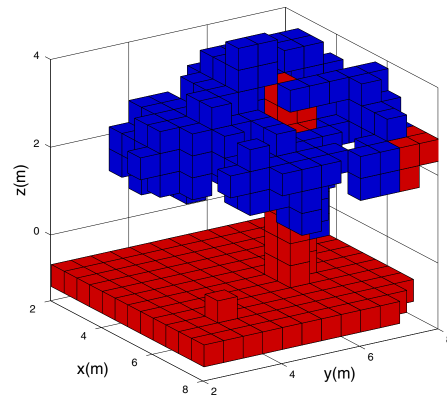

Fig. 6. Detail of cube classification for a tree from the first scene. In blue and red colors, collapsible and ground cubes, respectively.

averaging the height of the points that fall in 2D grid cells [2]. In the experiments, the DEM has been configured with a cell size of $0.5\,\mathrm{m}$. Second, fuzzy elevation maps (FEMs) represent elevation as a fuzzy surface, which filters sensor noise and interpolates missing data from small shadowed areas [22]. The FEM parameters can be identified from the point cloud by using ANFIS [23]. In particular, the computed FEMs have been defined by 190 rules with a total of 219 floating point parameters.

Figs. 7 and 8 show the elevation maps obtained from the raw point clouds for the first and second scenes, respectively. Red and blue colors indicate higher and lower elevations, respectively. The figures present most areas as non-reachable by a mobile robot starting on the current $(0, 0, -0.7\,\mathrm{m})$ position. Furthermore, these surfaces do not represent a feasible model of actual terrain surface. On the contrary, the elevation maps built using the point cloud given by the collapsible cubes method (see Figs. 9 and 10) aptly represent traversable areas while maintaining vertical structures.

V. CONCLUSIONS

This paper has proposed a simple processing of a leveled 3D point cloud that identifies and removes overhang points. The method uses efficient data structures to assign all points to coarse binary cubes. In the proposed procedure, an occupied cube is classified as collapsible when there is a vertical gap with respect to the ground that surpasses the height of an autonomous vehicle.

The method has been successfully applied to ground-based data from a 3D laser scanner on a mobile robot in outdoor environments. Computation times show an improvement with respect to a previous point-based solution to this problem. Furthermore, the resulting point clouds has been applied to build standard and fuzzy elevation maps (FEMs) of these outdoor scenes. Future work includes autonomous navigation tests with the Quadriga mobile robot based on local planned paths from the FEMs [24].

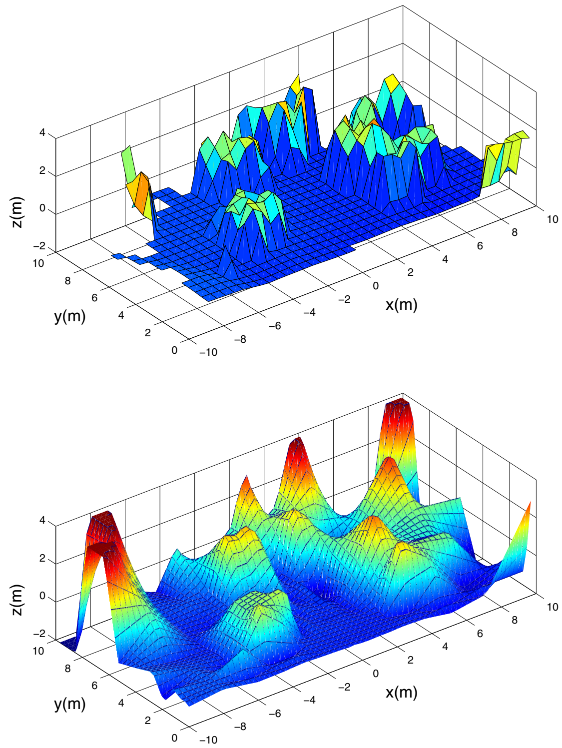

Fig. 7. DEM (top) and FEM (bottom) obtained from the raw scan of the first scene.

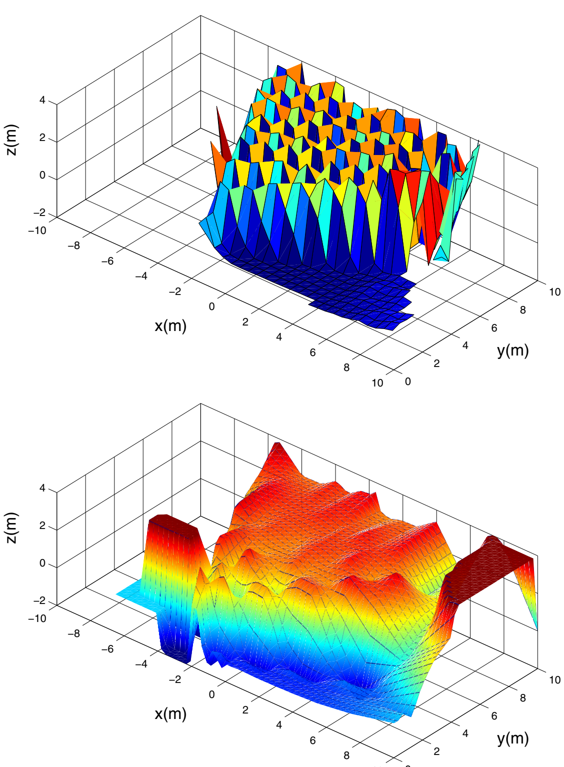

Fig. 8. DEM (top) and FEM (bottom) obtained from the raw scan of the second scene.

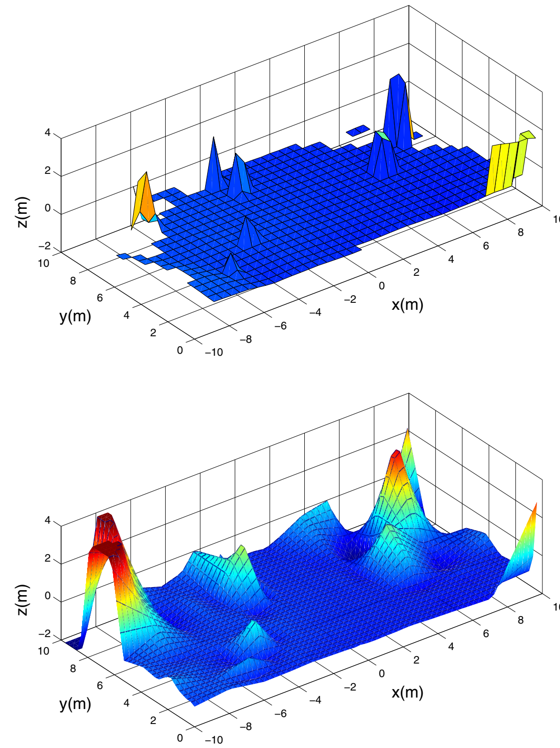

Fig. 9. DEM (top) and FEM (bottom) obtained from the scan without overhanging points of the first scene.

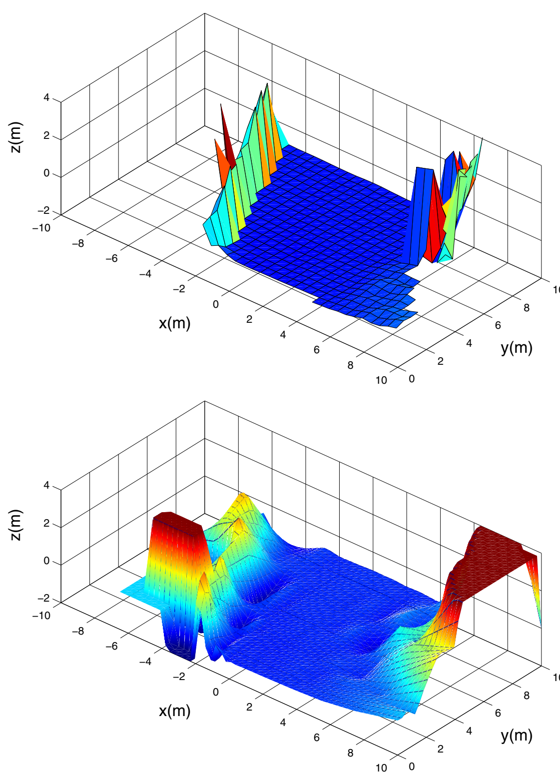

Fig. 10. DEM (top) and FEM (bottom) obtained from the scan without overhanging points of the second scene.

Acknowledgments

This work was partially supported by the Spanish CICYT project DPI 2011-22443 and the Andalusian project PE-2010 TEP-6101.

References

[1] A. Hornung, K. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[2] W. Burgard and M. Herbert, "World modeling," in *Handbook of Robotics*, B. Siciliano and O. Kratib, Eds. Springer, 2008, ch. 36.

[3] P. Pfaff, R. Triebel, and W. Burgard, "An efficient extension to elevation maps for outdoor terrain mapping and loop closing," *International Journal of Robotics Research*, vol. 26, no. 2, pp. 217–230, 2007.

[4] G. Ishigami, K. Nagatani, and K. Yoshida, "Path planning and evaluation for planetary rovers based on dynamic mobility index," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, USA, 2011, pp. 601–606.

[5] D. Gingras, T. Lamarche, J. L. Bedwani, and E. Dupuis, "Rough terrain reconstruction for rover motion planning," in *Proc. Canadian Conference on Computer and Robot Vision*, Ottawa, Canada, 2010, pp. 191 – 198.

[6] I. Rekleitis, J.-L. Bedwani, E. Dupuis, T. Lamarche, and P. Allard, "Autonomous over-the-horizon navigation using LIDAR data," *Autonomous Robots*, vol. 34, pp. 1–18, 2013.

[7] G. Reina, N. Giannoccaro, A. Messina, and A. Gentile, "Mobile robot perception using an inexpensive 3-D laser rangefinder," in *Proc. IEEE International Symposium on Industrial Electronics*, Bari, Italy, 2010, pp. 2809–2814.

[8] C. Rivadeneyra and M. Campbell, "Probabilistic multi-level maps from LIDAR data," *International Journal of Robotics Research*, vol. 30, no. 12, pp. 1508–1526, 2011.

[9] R. Triebel, P. Pfaff, and W. Burgard, "Multi-level surface maps for outdoor terrain mapping and loop closing," in *Proc. IEEE International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 2276–2282.

[10] R. Murai, T. Sakai, Y. Kitano, and Y. Honda, "Recognition of 3D dynamic environments for mobile robot by selective memory intake and release of data from 2D sensors," in *Proc. IEEE/SICE International Symposium on System Integration*, Fukuoka, Japan, 2012, pp. 621–628.

[11] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the DARPA grand challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[12] B. Douillard, J. Underwood, N. Melkumyan, S. Singh, S. Vasudevan, C. Brunner, and A. Quadros, "Hybrid elevation maps: 3D surface models for segmentation," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp. 1532–1538.

[13] J.-F. Lalonde, N. Vandapel, D. Huber, and M. Hebert, "Natural terrain classification using three-dimensional ladar data for ground robot mobility," *Journal of Field Robotics*, vol. 23, no. 10, pp. 839–861, 2006.

[14] A. Kelly, A. Stentz, O. Amidi, M. Bode, D. Bradley, A. Diaz-Calderon, M. Happold, H. Herman, R. Mandelbaum, T. Pilarski, P. Rander, S. Thayer, N. Vallidis, and R. Warner, "Toward reliable off road autonomous vehicles operating in challenging environments," *International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 449–483, 2006.

[15] D.-V. Nguyen, L. Kuhnert, T. Jiang, S. Thamke, and K.-D. Kuhnert, "Vegetation detection for outdoor automobile guidance," in *Proc. IEEE International Conference on Industrial Technology*, Auburn, USA, 2011, pp. 358–364.

[16] A. Cappalunga, S. Cattani, A. Broggi, M. McDaniel, and S. Dutta, "Real time 3D terrain elevation mapping using ants optimization algorithm and stereo vision," in *Proc. IEEE Intelligent Vehicles Symposium*, 2010, pp. 902–909.

[17] A. Broggi, E. Cardarelli, S. Cattani, and M. Sabbatelli, "Terrain mapping for off-road autonomous ground vehicles using rational B-Spline surfaces and stereo vision," in *Proc. IEEE Intelligent Vehicles Symposium*, 2013, pp. 648–653.

[18] J.-F. Lalonde, N. Vandapel, and M. Hebert, "Data structures for efficient dynamic processing in 3-D," *International Journal of Robotics Research*, vol. 26, no. 8, pp. 777–796, 2007.

[19] J.-L. Martínez, A. J. Reina, A. Mandow, and J. Morales, "3D registration of laser range scenes by coincidence of coarse binary cubes," *Machine Vision and Applications*, vol. 23, pp. 857 –867, 2012.

[20] J. Morales, J.-L. Martínez, A. Mandow, A. Pequeño-Boter, and A. García-Cerezo, "Design and development of a fast and precise low cost 3D laser rangefinder," in *Proc. IEEE International Conference on Mechatronics*, Istanbul, Turkey, 2011, pp. 621–626.

[21] J. Morales, J. L. Martínez, A. Mandow, A. Pequeño-Boter, and A. García-Cerezo, "Simplified power consumption modeling and identification for wheeled skid-steer robotic vehicles on hard horizontal ground," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp. 4769–4774.

[22] A. Mandow, T.-J. Cantador, A. García-Cerezo, A.-J. Reina, J.-L. Martínez, and J. Morales, "Fuzzy modeling of natural terrain elevation from a 3D scanner point cloud," in *Proc. 7th IEEE International Symposium on Intelligent Signal Processing*, Floriana, Malta, 2011.

[23] J.-S. R. Jang, "ANFIS: Adaptative-network-based fuzzy inference system," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, 1993.

[24] J.-L. Martínez, A. Mandow, A. Reina, T.-J. Cantador, J. Morales, and A. García-Cerezo, "Navigability analysis of natural terrains with fuzzy elevation maps from ground-based 3D range scans," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013.