

# 3D traversability awareness for rough terrain mobile robots

Mauro Bellone, Giulio Reina, Nicola Ivan Giannoccaro and Luigi Spedicato

Department of Engineering for Innovation, University of Salento, Lecce, Italy

## Abstract

**Purpose** – This research aims to address the issue of safe navigation for autonomous vehicles in highly challenging outdoor environments. Indeed, robust navigation of autonomous mobile robots over long distances requires advanced perception means for terrain traversability assessment.

**Design/methodology/approach** – The use of visual systems may represent an efficient solution. This paper discusses recent findings in terrain traversability analysis from RGB-D images. In this context, the concept of point as described only by its Cartesian coordinates is reinterpreted in terms of local description. As a result, a novel descriptor for inferring the traversability of a terrain through its 3D representation, referred to as the unevenness point descriptor (UPD), is conceived. This descriptor features robustness and simplicity.

**Findings** – The UPD-based algorithm shows robust terrain perception capabilities in both indoor and outdoor environment. The algorithm is able to detect obstacles and terrain irregularities. The system performance is validated in field experiments in both indoor and outdoor environments.

**Research limitations/implications** – The UPD enhances the interpretation of 3D scene to improve the ambient awareness of unmanned vehicles. The larger implications of this method reside in its applicability for path planning purposes.

**Originality/value** – This paper describes a visual algorithm for traversability assessment based on normal vectors analysis. The algorithm is simple and efficient providing fast real-time implementation, since the UPD does not require any data processing or previously generated digital elevation map to classify the scene. Moreover, it defines a local descriptor, which can be of general value for segmentation purposes of 3D point clouds and allows the underlining geometric pattern associated with each single 3D point to be fully captured and difficult scenarios to be correctly handled.

**Keywords** Mobile robots, Local descriptor, Point feature representation, Unevenness point descriptor

**Paper type** Research paper

## Erratum

*It has come to the attention of Emerald Group Publishing that the article “3D traversability awareness for rough terrain mobile robots”, published in Sensor Review, Vol. 34 No. 2, pp. 220–232, contained a number of errors. This occurred due to an error in the editorial process. Emerald sincerely apologises to the authors and the readers for this inconvenience. This has been corrected in the online version of the article.*

## I. Introduction

Research in mobile robotics aims at developing technologies enabling vehicles to travel longer distances with limited human supervision (Reina *et al.*, 2010). If autonomous vehicles could drive reliably and robustly through unknown terrain toward a given location, the implications would be of great importance for many applications, including mining, earthmoving, agriculture, search and rescue, planetary exploration, and, in general, driverless vehicles. Although autonomous navigation has inspired decades of research, it still remains an open and active field of investigation. One of the critical challenges is accurate and robust perception of the scene to perform many important tasks, including

environment segmentation, classification, mapping and identification of safe traversable regions of terrain (Reina *et al.*, 2012). New interest in the field has been raised by the introduction of the next generation of stereo and depth cameras that produce accurate 3D representations of the environment in the form of 3D point clouds. Given a raw 3D point cloud, the degree of perception depends on the ability to find models within a scene and matching local regions. Researchers usually think of a 3D point as defined just by its Cartesian coordinates  $(x, y, z)$  according to the 3D meaning of the Euclidean metric. However, if only the Cartesian coordinates of a 3D point are taken into account, the problem of object recognition may be ill-posed. Therefore, in the context of perception, the concept of “local descriptor” (Rusu, 2009; Rusu *et al.*, 2010; Pauly and Kobbelt, 2002) should be emphasized, in which “rich” 3D data are considered to perform high-level tasks.

This research addresses the issue of safe navigation for autonomous vehicles using 3D vision sensors. Specifically, a new approach for terrain traversability assessment using a 3D descriptor, referred to as the unevenness point descriptor (UPD), is presented. This approach is based on the concept of normal vector to a local surface and it uses principal component analysis (PCA) for fast and accurate scene interpretation. By observing the distribution of normal vector direction, it is possible to interpret the scene giving a traversability index point

The current issue and full text archive of this journal is available at [www.emeraldinsight.com/0260-2288.htm](http://www.emeraldinsight.com/0260-2288.htm)



Sensor Review  
34/2 (2014) 220–232  
© Emerald Group Publishing Limited [ISSN 0260-2288]  
[DOI 10.1108/SR-03-2013-644]

The financial support of the ERA-NET ICT-AGRI through the grant Ambient Awareness for Autonomous Agricultural Vehicles (QUAD-AV) is gratefully acknowledged. Further thanks to Mrs Rita Bennett and Mrs Sara Bianco for their productive contribution in language revision.

by point and to enable safe driving. The main advantages of the proposed algorithm are:

- It is simple and computationally efficient, providing real-time implementation.
- It lends itself very well to the novel imaging technologies based on 3D point cloud processing.
- It can be successfully applied for outdoor scene interpretation handling critical scenarios including slopes and ramps.
- It provides as output a traversability map that can be directly applied to most grid-based path planners.

The system is demonstrated in field experiments showing its effectiveness for scene interpretation and automatic obstacle detection in both indoor and outdoor environments.

The paper is organized as follows. In Section II, a thorough survey of previous work in this area can be found. Section III revolves around the theory behind the concepts of local descriptor and PCA. In Section IV, the UPD descriptor is presented and discussed in detail. Section V describes the algorithm to generate the UPD and some preliminary results are given using simulated data. The application of the UPD-based analysis for mobile robot applications is experimentally validated in Section VI, in which its effectiveness for terrain traversability is demonstrated. Relevant conclusions are drawn in Section VII.

## II. Related works

Past research addressed the problem of terrain analysis following different approaches. Proposed solutions change according to available sensors and data, as presented by Papadakis (2013), wherein an overview of terrain traversability analysis methods for unmanned vehicles can be found. Most of the methods are based on visual information, as in Milella *et al.* (2006), in which two visual algorithms are presented. The first approach concerns 6DoF ego-motion estimation, whereas the latter estimates wheel sinkage in sandy soil. Using visual information (Braun *et al.*, 2008), a stereo-vision-based terrain traversability estimation method for off-road mobile robots is presented. The authors use normal vectors in order to estimate the inclination of a sector in an elevation map; moreover, they classify as non-traversable some regions where the inclination is higher than the vehicle is capable of scaling. However, the high computational cost of their approach would make it difficult to use in real-time navigation tasks. A different method was studied by Dongshin *et al.* (2007), in which the authors describe a method for terrain classification using point clouds data obtained by a stereovision. They propose the use of superpixels as the visual primitives for traversability estimation using a learning algorithm. On the other hand, 3D depth sensors including LIDARS or depth cameras have also been investigated. As an example, Vandapel *et al.* (2004) acquire information about terrain using a LIDAR and using local 3D point statistics segment it into three classes: clutter to capture grass and tree canopy, linear to capture thin objects like wires or tree branches, and finally surface to capture solid objects like ground terrain surface, rocks or tree trunks. As a further example, Larson *et al.* (2011) discuss a real-time approach to analyze the traversability of off-road terrain for unmanned ground vehicles considering positive and negative obstacles through elevation information acquired by a LIDAR. A different approach to traversability estimation based on

learning algorithm and applied to planetary rovers was described by Howard *et al.* (2007). Their implemented learning algorithm was able to classify terrain by learning from 3D geometry information and by learning from proprioception. In the same field, a study of multi-sensor terrain classification for planetary rovers in Mars-like environments is investigated by Halatci *et al.* (2007), wherein the authors propose two classification algorithms based on Bayesian fusion and meta-classifier fusion. Other works use different probabilistic methodologies for terrain classification; an additional example can be found in Kim *et al.* (2010), in which the authors build traversability maps on camera images using a Bayesian approach. A different approach consists of defining a digital elevation map and assigning two indices to each of its cells: the roughness and the inclination index (Howard and Saraji, 2001; Osari, 2003; Seraji, 1999). The former is defined as the variance of the elevation values in a specific region of the environment, whereas the latter can be obtained as the average angle of adjacent elevation values with respect to their neighbors (Rohmer *et al.*, 2010). Merging the research novelties in data processing and 3D sensors, new approaches were presented. As an example, Du Pont *et al.* (2008), classify terrains using PCA theory and vibration sensor information. As a matter of fact, 3D data processing for terrain analysis has gained interest in research as expressed by Silver *et al.* (2006), in which the researchers employ laser range finders or overhead cameras to capture surface 3D point cloud data and PCA analysis for terrain evaluation. Although data processing can provide calculation and analysis, for the practical robotic applications it is required to classify data and interpret it. A self-learning ground classifier based on geometric and shape features extracted from radar and stereo data are presented by Reina and Milella (2012) and Reina *et al.* (2012), respectively. A relevant example of practical application is presented by Neuhaus *et al.* (2009), in which the authors propose a new method for navigation in an unstructured environment using a grid-based approach.

The strengths of this method in comparison to previous approaches are:

- The definition of a local descriptor that can be of general value for segmentation purposes of 3D point clouds. It can also be directly applied to different specific tasks, including terrain analysis, traversability assessment, and path planning.
- The use of a descriptor in contrast to a single feature or a set of features, which allows the underlying geometric pattern associated with each single 3D point to be fully captured and difficult scenarios including ramps, edges, and negative obstacles to be correctly handled. It should also be added that, the estimation of the UPD in a query point takes into account the geometric properties of not only the points located inside the neighborhood, but also those located in its vicinity, providing accuracy and robustness to the system.
- The low-computational burden, since UPD does not require any data processing or previously generated digital elevation map to classify the scene, as it can be directly applied to raw 3D point clouds.

## III. Theoretical background

This work relies on PCA-based surface analysis for the extraction of local information from a point cloud. In order to

provide the necessary background for those unfamiliar with the topic, this section recalls two theoretical tools: the concept of local descriptor and surface analysis through PCA.

**A. Local descriptors**

Perception can be defined as the problem of describing a point through the definition of an appropriate set of salient local characteristics. Figure 1 shows an illustrative example that will help to clarify this concept. In particular,  $p_1$  can be seen as a single point, yet it is the intersection of three perpendicular planes representing the sides of a cube. Similarly,  $p_2$  can also be seen as the intersection between two perpendicular planes. Therefore,  $p_1$  and  $p_2$  can be described by their Cartesian coordinates augmented with the attribute of “vertex” or “edge”, respectively, according to their local characteristics or features. The set of characteristics used to describe a point defines a “local descriptor”.

The concept of local descriptor can be mathematically derived. Let  $I$  be a point cloud, i.e. a set of  $n$  points defined by their Cartesian coordinates  $I = \{p_i \in \mathbb{R}^3, \quad i = 1, 2, \dots, n\}$ , and let  $p_q$  be a given point defined as the query point. The neighborhood of  $p_q$  in  $I$  can be defined as the set of points such that:

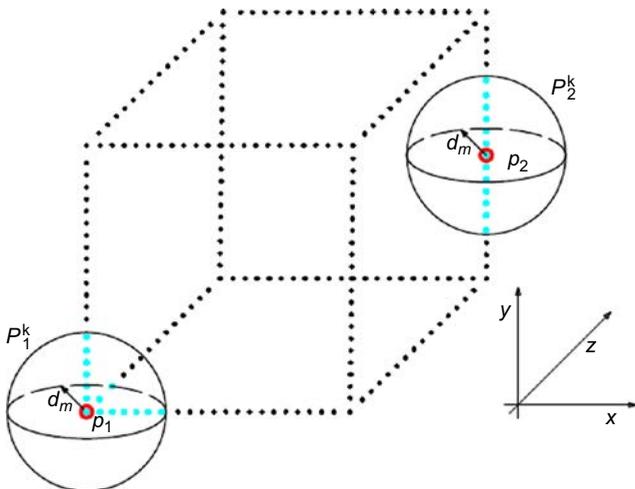
$$p_q^k = \{p_i^k \in I \subset \mathbb{R}^2 : |p_i^k - p_q| \leq d_m \quad \forall i = 1, 2, \dots, k\} \quad (1)$$

where  $d_m$ , so-defined as the search radius, is the maximum distance between  $p_q$  and each neighbor,  $k$  the number of neighbors of  $p_q$ , and  $|\cdot|$  a generic norm (without loss of generality, it is possible to refer to the Euclidean distance). As an illustrative example, we can refer to Figure 1, where the neighborhood of the point  $p_1$  is represented by a sphere of center  $p_1$  and radius  $d_m$ , and its neighbors are marked by cyan dots. A local descriptor of  $p_q$  can be defined as the vector function  $F$  that describes the information content of  $P_q^k$  according to a specific feature set:

$$F(p_q, P_q^k) = \{x_1^q, x_2^q, \dots, x_n^q\}, \quad (2)$$

where  $x_j^q$  is the  $j$ th dimension of the descriptor. By comparing the local descriptors of two points, namely,  $p_1$  and  $p_2$ , it is possible to estimate their differences. Let  $\Gamma$  be the measure of similarity

**Figure 1** Points in the scene can be defined by their Cartesian coordinates augmented with their attributes or features that describe their local characteristics



between  $p_1$  and  $p_2$ , with their associated descriptors  $F_1$  and  $F_2$ , and let  $d$  be their distance:

$$\Gamma = d(F_1, F_2). \quad (3)$$

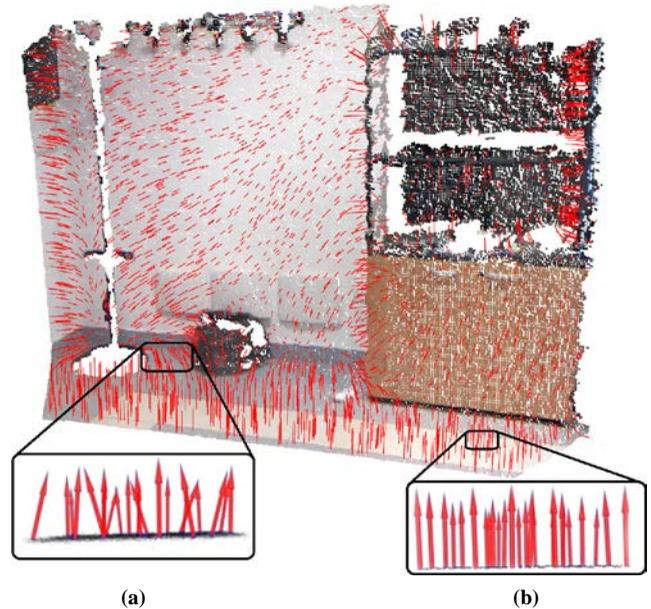
Then,  $d$  is a scalar value and can be considered as the degree of similarity between the given points. If  $d \rightarrow 0$  the points can be considered similar according to the specific feature set. Conversely, if  $d$  increases the points will have different properties. It is important to note that the effectiveness of the descriptor is given by its ability to differentiate points in the presence of rigid transformations, noise, sampling variations, changes in scale or illumination.

**B. Surface analysis**

A common method for surface analysis of 3D point clouds, is based on PCA. Numerous variations are also proposed as the PCA generalization (Kriegel *et al.*, 2008). Specifically, PCA provides an efficient method for the estimation of the normal vector to a surface. As an example, Figure 2 shows a sample RGB-D image acquired by a Microsoft Kinect (2010) sensor and processed using point cloud library (PCL; Rusu and Cousins, 2011), the normal vectors to local surfaces obtained from PCA are marked as red arrows. The distribution of the normal vector direction is rather uniform for smooth floors, see inset (b), whereas different directions can be observed in the vicinity of objects, see inset (a). This intuitively suggests a way to measure the smoothness of a surface based on the direction distribution of normal vectors, as exploited by the novel UPD, introduced later in the paper.

The problem of determining the surface inclination in a specific reference frame, can be seen as the problem of determining the normal vector of the tangent plane to the surface at the query point (Rusu, 2009). Although many different normal estimation methods exist, the simplest one is based on the first order 3D plane fitting. The tangent plane can

**Figure 2** Normal vector-based surface analysis of a RGB-D image



**Notes:** (a) Typical distribution of normal vectors found in an irregular region; (b) typical distribution of normal vectors in a regular region

be obtained as a least square fitting in  $P_q^k$ . A generic plane can be represented as a point  $x$  and its normal vector  $\vec{n}$ ; the distance between the generic point  $p_i \in P_q^k$  and the plane is defined as  $l_i = (p_i - x)\vec{n}$ . If  $x$  is the centroid of  $P_q^k$ , the solution for  $\vec{n}$  can be obtained by analysing the Eigenvalues and eigenvectors of the covariance matrix  $C \in \mathfrak{R}^{(3 \times 3)}$  of  $P_q^k$ , expressed by:

$$C = \frac{1}{k} \sum_{i=1}^k \xi_i (p_i - x)(p_i - x)^T \quad (4)$$

$$C \cdot \vec{v}_j = \lambda_j \cdot \vec{v}_j$$

The term  $\xi_i$  is the possible weight for  $p_i$ ,  $x$  is the centroid of  $P_q^k$  that does not necessarily coincide with  $p_q$ , and it is usually considered as unitary. The covariance matrix  $C$  is symmetric, positive-definitive, and its Eigenvalues are real. The eigenvectors  $v_j$  are an orthogonal subspace, and they are the principal components of  $P_q^k$ . Let us suppose that the Eigenvalues of  $C$  are  $0 \leq \lambda_0 \leq \lambda_1 \leq \lambda_2$ , then the eigenvector of the smallest Eigenvalue  $\lambda_0$  represents the approximation of the normal vector  $\vec{n} = (n_x, n_y, n_z)$  or its opposite  $-\vec{n}$ . More details about this formulation can be found in Rusu (2009).

One should note that the sign ambiguity of the normal vector is not analytically solvable, yet it is generally estimated through the analysis of the other eigenvectors that complete the Eigenspace. As a result, each surface will have two normal vectors, one pointing upwards and one pointing downwards. Here, conventionally all normal vectors will be considered pointing upwards.

#### IV. Unevenness point descriptor

Normal vectors can be used for terrain analysis purposes. Non-traversable obstacles or high irregular ground can be detected and avoided by an autonomous vehicle along its path toward a target in both indoor and outdoor scenarios. In this section, a new approach for terrain evaluation is proposed using a new point descriptor referred to as the UPD.

The idea behind the UPD is to have a simple compound measure that takes into account both terrain irregularity and inclination.

Let us consider the query point  $p_q$ , and its neighborhood  $P_q^k$ , computed as defined in Section II. Given the  $\vec{n}_i$  normal vectors, with  $i = 1, 2, \dots, k$ , where  $k$  is the number of neighbours, it is possible to compute their vector sum:

$$\vec{r}_k^q = \vec{n}_1 + \vec{n}_2 + \dots + \vec{n}_k = \sum_{i=1}^k \vec{n}_i. \quad (5)$$

Then, the UPD  $F_U$  in  $p_q$ , can be defined as:

$$F_U(p_q, P^k) = \left\{ r_x^q, r_y^q, r_z^q, \zeta_k^q \right\}, \quad (6)$$

where  $r_x^q, r_y^q, r_z^q$  are the scalar components of  $\vec{r}_k^q$ , and  $\zeta_k^q$  is defined by the following equation (7). The components of  $\vec{r}_k^q$  provide information about the global direction of the local surface in the sensor reference frame. On the other hand, the parameter  $\zeta_k^q$  can be interpreted as a local inverse ‘‘unevenness index’’, since it assesses the degree of local roughness. It depends on the distribution of the direction of the normal vectors in the neighborhood and it is given by:

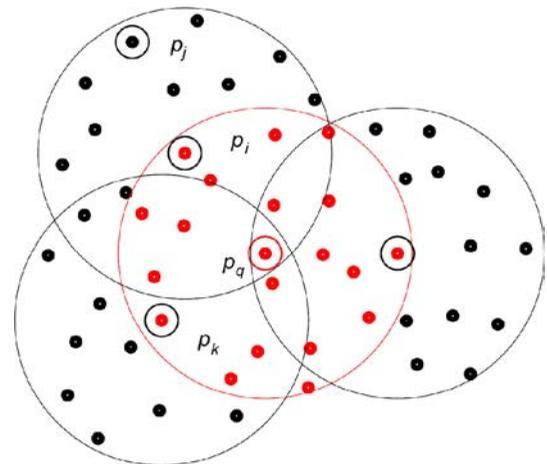
$$\zeta_k^q = \frac{\|\vec{r}_k^q\|}{k}, \quad (7)$$

where  $\vec{r}_k^q$  has been defined in equation (5), and  $k$  is the number of neighbors in  $P_q^k$ . By dividing by  $k$ ,  $\zeta_k^q$  is normalized and neighborhoods, including a different number of points, can be compared to each other.

The main advantages of this descriptor reside in its simplicity and robustness for traversability evaluation. Moreover, the UPD can be easily adapted to the specific task of the robot by appropriately setting the neighborhood size, i.e.  $d_m$  in equation (1). In practice, its value will be fixed at the beginning of the operations based on the robot geometric size (Bellone *et al.*, 2013). However, the performance of the system is greatly affected by the search radius. If  $d_m$  is too large, all normal vectors will influence each other producing poor results and making it difficult to detect the exact location of the discontinuity. As a general rule, the larger the search radius, the wider the portion of the environment to consider for the estimation of the UPD in the given query point. Therefore, all obstacles within a distance of  $d_m$  from this point will affect the index estimation. This consideration in conjunction with the knowledge of the vehicle geometric size and the field of view of the sensor, suggest a possible approach for the optimal choice of  $d_m$ .

Now, one should note that the vectors influencing the UPD belong to the points in the neighborhood of the query point  $p_q$ , although each normal vector is computed including a different set of points. Hence, the informative content of UPD includes a piece of information not only about the points located inside the neighborhood, but also about those which are located in its vicinity. In order to clarify this concept, Figure 3 shows a representation of the neighborhoods influencing directly and indirectly the UPD value in  $p_q$ . In particular, the red circumference denotes the neighborhood of  $p_q$ , whereas the black circumferences denote the neighborhoods of three example points. The point  $p_j$  belongs to the neighborhood of  $p_i$ , while  $p_i$  belongs to the neighborhood of  $p_q$ . Hence,  $p_i$  directly influence the UPD value in  $p_q$ , whereas  $p_j$  indirectly influences it, even though it does belong to neighborhood of  $p_q$ . From a practical point of view, the normal vector of  $p_i$  is calculated including, in the covariance matrix, the informative content of the point  $p_j$ .

Figure 3 Normal vector-based surface analysis of a RGB-D image



**Note:** The neighborhood of  $p_q$  is directly influenced by the point denoted in red, whereas the points denoted in black influence the value of the normal vector of the neighbors of  $p_q$

### A. Remarks

This section underlines two remarks regarding the UPD. The first concerns the evaluation of its consistency. Essentially, we are interested in knowing when the robot can trust UPD value. Its consistency is mainly influenced by two factors:

- 1 the density of the cloud; and
- 2 the stability of the surface.

Both are related to the information content of the neighborhood. The second aspect concerns the orientation of surfaces representing possible obstacles and, consequently, their interpretation.

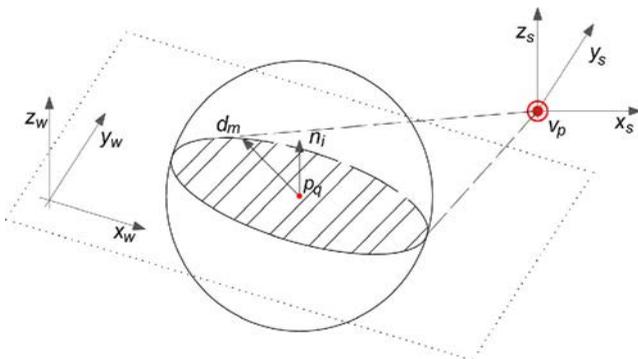
In general, 3D point clouds are composed by dense and not-dense areas due to sensor limitations (e.g. occluded areas or pixels at long distances). This affects the local density of the point cloud. With the aim to describe the local density of a point cloud, let us define the scalar value  $\rho_v$  as “volumetric discrete density” of a point cloud, given by the number of points in the volume unit. In the same way, it is possible to define the “surface discrete density”  $\rho_{as}$  as the number of points located in a surface unit.

Let us consider  $\rho_v$  as the volumetric discrete density as defined above, the maximum number of points in a neighborhood  $m_p$  can be expressed by the product between the volumetric density and the volume of the sphere with radius  $d_m$ :

$$m_p = \rho_v \left( \frac{2}{3} \pi d_m^3 \right), \quad (8)$$

where  $d_m$  is the search radius of the neighborhood as defined in equation (1). However, the formulation in equation (8) does not take into account that, from a practical point of view, there is no possibility of having the volume completely full of points. Indeed, the laser beam originated from the sensor impacts an object generating a set of points describing the shape of its surface. In order to better clarify this concept, let us consider Figure 4, where  $v_p$  is the “viewpoint” of the sensor. Figure 4 describes a typical neighborhood, supposed to be a sphere with radius  $d_m$ , of a generic point  $p_q$  belonging to a plane. As every neighborhood, it can include only the information of a surface portion. Even though the surface is irregular, the neighborhood will describe only a portion of it. Therefore, the estimated number of points in a neighborhood is given, with reasonable accuracy, by the number of points in the intersection between a

**Figure 4** Representation of surface acquired from a sensor



**Note:** The  $v_p$  is the viewpoint,  $(x_s, y_s, z_s)$  is the sensor reference frame,  $(x_w, y_w, z_w)$  is the surface reference frame,  $p_q$  is the query point with its normal vector  $n_i$  and  $d_m$  is the radius of the sphere representing the neighborhood

planar surface and the sphere supposed to be the shape of the neighborhood. Such intersection is a circumference. Moreover, if the query point belongs to the intersecting surface, and it is considered as the centroid of the neighborhood, then the intersecting circumference will have radius  $d_m$ . Given the above, it is possible to rewrite equation (8) as the product between the surface density  $\rho_a$  and the area of the circumference with radius  $d_m$ :

$$m_e = \rho_a (2\pi d_m^2). \quad (9)$$

The parameter  $m_e$  is defined as the expected number of points in the neighborhood. One should note that, in equation (9) the surface density of the cloud  $\rho_{as}$  is considered instead of  $\rho_v$ . Given the above, the ratio between the number of points in the neighborhood  $k$ , see equation (1), and the number of expected points in the same neighborhood  $m_e$ , see equation (9), is given by:

$$c = k/m_e. \quad (10)$$

From the equation (10) it should be noted that:

- If  $k \approx m_e$  then  $c \approx 1$  and the neighborhood has exactly the expected number of points.
- If  $k \ll m_e$  then  $c \ll 1$  and the neighborhood has less point than expected. As a result, the neighborhood may not have enough information for estimating the UPD.
- If  $k \gg m_e$  then  $c \gg 1$  and the neighborhood has more points than expected. Hence, there is redundancy of information in the neighborhood.

Generally,  $c \ll 1$  in the vicinity of an object boundary, where there is a loss of information due to the surface variation. This happens often in the case of the proximity of border regions, where generally there is not enough information to fully describe the terrain. Whereas,  $c \gg 1$  in the case of possible reflections that produce noise and false points in the laser acquisition.

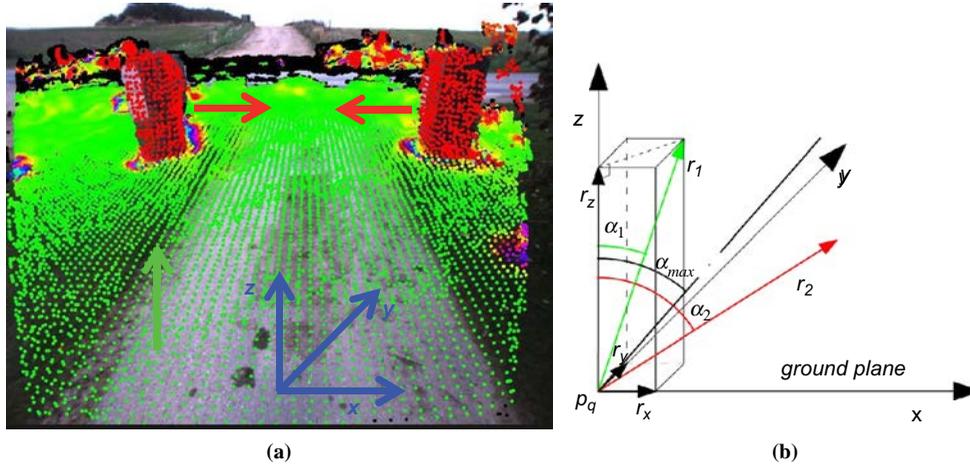
The last remark concerning the UPD is the orientation analysis. From a mathematical point of view, vertical and horizontal surfaces might be represented by the same value of the unevenness index, since it provides regularity information only, regardless of the surface inclination. Although the horizontal surface can be traversed, the vertical one typically describes obstacles. In Figure 5(a), an explicative example is shown. Here, the ground reference frame is marked as blue arrows and a typical vector belonging to the ground is marked as a green arrow. One should note that a typical ground normal vector should be aligned with the  $z$ -axis. In the same Figure 5(a), the red arrows mark vectors belonging to objects. As shown, their orientation is typically misaligned with respect to  $z$ -axis. Hence, given the query point  $p_q$  and its  $r$ -vector, the angle  $\alpha_q$  between the  $z$ -axis and the  $r$ -vector can be obtained as:

$$\alpha_q = \arccos \left( \frac{r_z^q}{\|r_q\|} \right). \quad (11)$$

The orientation analysis can be translated into a simple condition where a point can be considered as non-traversable if  $\alpha_q > \alpha_{max}$ , where  $\alpha_{max}$  is a user choice and it should be taken according to the vehicle capability.

In Figure 5(b) a graphical explanation can be seen. The green vector  $r_1$  can be supposed to be ground, whereas the red vector  $r_2$  may belong to a non-traversable surface. Their angle are, respectively,  $\alpha_1$  and  $\alpha_2$ , and they are supposed to be  $\alpha_1 < \alpha_{max} < \alpha_2$ .

Figure 5 Orientation analysis



**Notes:** (a) The orientation analysis superimposed on the camera image: the green arrow marks a typical normal vector belonging to the ground, whereas the red arrows mark typical normal vectors belonging to obstacles; the ground reference frame is marked in blue; (b) geometric interpretation of the orientation analysis based on the UPD

## V. Algorithm description

In this section, the algorithm for UPD will be thoroughly described. Computationally and numerically speaking, the factors that mostly affect the performance of the algorithm are the total number of points in the 3D image and the dimension of the neighborhood  $P^k$ . Specifically, the computational burden of the algorithm increases with the dimension of  $P^k$ , which is related to the search radius  $d_m$ . The algorithm might be seen as divided into two main stages: the normal vector estimation and the UPD analysis (Algorithm 1). The former includes the steps from 1 to 6, where the algorithm computes the normal vector for each point in the 3D image, whereas the latter includes steps from points 7 to 12, in which the algorithm performs the UPD analysis.

The algorithm for the estimation of the UPD can be summarized as follows:

### Algorithm 1. Pseudocode of the UPD

Normal vector estimation:

1. Point cloud acquisition, filtering and downsampling
2. **For** all  $p_i$  **do**
3.  $P_k^i$  selection, equation (1)
4. Compute the covariance matrix in  $P_k^i$ , equation (4)
5. Compute normal vector of  $p_i$  as the eigenvector of  $\lambda_0$  in equation (4)
6. **End**

UPD analysis:

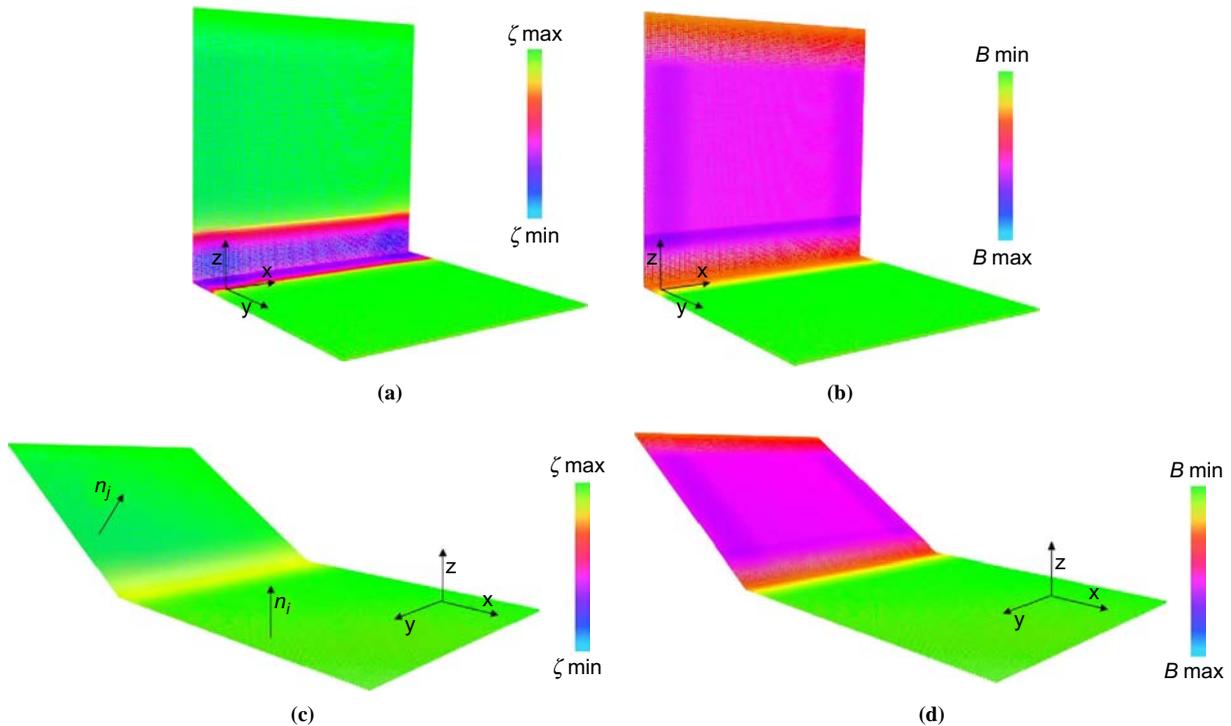
7. **For** all  $p_i$  **do**
8.  $p_k^i$  selection equation (1)
9. Compute equation (10)
10. **if**  $c > c_{min}$
11. Compute the vector sum equation (5)
12. **if**  $\alpha_i < \alpha_{max}$  the point  $p_i$  might be traversable
13. Compute Unevenness Index equation (7)
14. **else**, the point  $p_i$  is not traversable
15. **else**, no correct estimation of the normal vector in  $p_i$
16. **End**

The first stage regarding normal vectors estimation is simple and well known. Here, we will focus our attention on the second stage concerning the UPD analysis. This part of the algorithm was optimized for avoiding unnecessary calculations. First of all, it is necessary to check whether the selected neighborhood contains useful information, using equation (10) and defining a minimum threshold  $c_{min}$ . If  $c < c_{min}$ , it is not possible to have a correct estimation of the normal vectors. If the condition is verified, then the normal vector estimated in  $p_i$  can be considered trustworthy, otherwise the estimation of the normal vector is considered uncertain and the associated UPD calculation is not performed. After this, equation (5) provides the vector sum of all normal vectors in the neighborhood and it is possible to check and classify obstacles using the condition on the line 12 of Algorithm 1, i.e.  $\alpha_i < \alpha_{max}$ . Finally, using equation (7) the unevenness index associated with  $p_i$  can be obtained.

### A. Comparison with the literature

In order to compare the proposed approach with existing methods, UPD-based analysis and roughness analysis (Rohmer *et al.*, 2010), are implemented using simulated data. The use of simulated data provides a reliable comparison framework. The simulations include two different data sets. In the first data set, the presence of a 90° corner is generated by intersecting two orthogonal planes whose dimensions are of 1 × 1 m with a resolution of 0.01 m. Point clouds are automatically generated by a C++ code and represented by a grid with points 0.01 m apart. In the second data set, a 30° ramp is simulated. Attention is given to the unevenness index, i.e. the fourth component of the UPD, see equation (7). In general, high values of  $\zeta_k^q$  suggest regularity, low values indicate irregularity, as reflected by the color scale associated with Figure 6.

Figure 6(a) shows the results obtained applying the unevenness index to the 90° corner. The search radius used in this simulation is 0.1 m. The UPD analysis results in low values of  $\zeta_k^q$  in proximity of the corner that is correctly detected as an irregular region. On the other hand, Figure 6(b) shows

**Figure 6** Comparison of the UPD with the roughness index for simulated surface

**Notes:** (a)-(c) Results obtained from the UPD-based analysis; (b)-(d) results obtained from the roughness index; the UPD labels both ground and ramp as traversable, whereas the roughness index provides different scores  $B$  for the ground ( $B = B_{min}$ ) and the ramp ( $B = B_{max}$ )

the same scenario analyzed by applying the roughness index  $B$ , i.e. the standard deviation of the terrain elevation over a local surface, which is assumed to be equal to the neighborhood of the query point in this analysis. By considering the terrain elevation only, the roughness index erroneously marks the whole surface as irregular. In Figure 6(c) and (d), the UPD is compared with the roughness index for a scenario featuring a ramp with a slope of  $30^\circ$ . Figure 6(c) shows the results obtained from the UPD-based approach. As expected, the lower border of the ramp is flagged as irregular, whereas the ramp itself is correctly labeled as traversable. One should note that, the lower border of the ramp is flagged as irregular, yet the unevenness index is higher than in Figure 6(a). As a result, the UPD is able to distinguish between different levels of irregularity. This represents one advantage of the UPD-based analysis. In Figure 6(d), the same scene is analyzed applying the roughness index. As in Figure 6(b), by relying only on terrain elevation, the ramp is misclassified as non-traversable by the roughness index. In contrast, the UPD approach is based on the analysis of the local normal vectors providing a similar score for the ground and the ramp that are both classified as traversable.

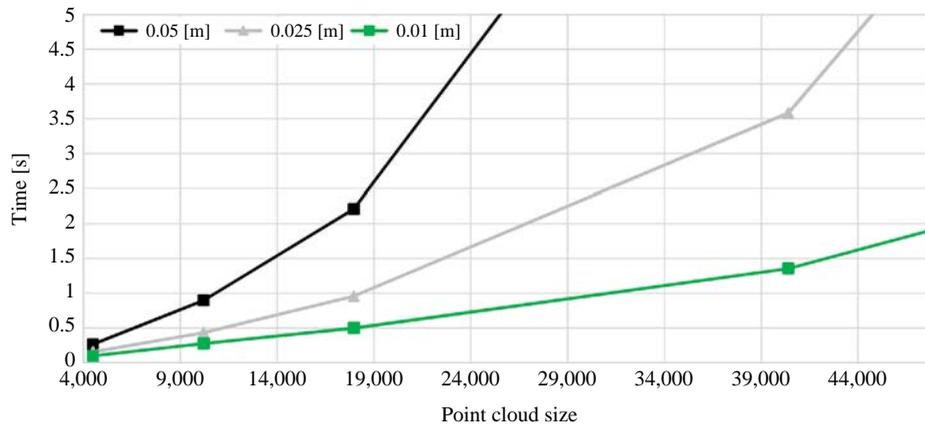
### B. Computational performance

The algorithm was implemented in C++ including PCL for fast 3D data elaboration. The tests were performed on an ASUS-EeePC-1215N. The choice of using a low performance computer for the tests derives from the practical application on robots featuring low performance on-board computers.

The computational performance of the UPD-based algorithm is affected by various factors, such as point cloud size, resolution and data denseness. Moreover, the search radius  $d_m$ , introduced in equation (1), affects the performance of the algorithm and it should be chosen according to the specific application and robot. In order to show the effect of the parameters tightly related to the UPD, the computational performance was evaluated on simulated data including a single point cloud. The simulated point cloud is a  $1 \times 1$  m regular plane, numerically generated and perfectly homogenous. Table I lists the results of the algorithm. To evaluate the performance in different conditions, the plane resolution was progressively increased with the point cloud size, ranging from 4,489 to 161,225 points. Then, the execution time was measured, by a C++ routine, varying the search radius from 0.01 m to 0.05 m. The values in Table I are also plotted in Figure 7, in which a graph of the computational performance is shown. The computational time for the case  $d_m = 0.01$  m is plotted against the point cloud size using a green line, whereas

**Table I** Computational performance

Point clouds size	$d_m = 0.01$ (m)	$d_m = 0.025$ (m)	$d_m = 0.05$ (m)
	Execution time (s)		
4,489	0.096	0.153	0.263
10,201	0.274	0.431	0.895
17,956	0.497	0.957	2.209
40,401	1.353	3.586	10.501
160,801	10.394	42.859	161.225

**Figure 7** Computational performance graph of the results collected in Table I

**Note:** The computational time is evaluated against the point cloud size for three search radius values, 0.05, 0.025 and 0.01 m

grey and black lines show the cases of  $d_m = 0.025$  m and  $d_m = 0.05$  m, respectively. At a constant point cloud size, the larger the search radius, the higher the computational time. However, the use of a large search radius leads to poor results for scene interpretation due to over-fitting, i.e. too many points included in the estimation of UPD.

In summary, the UPD algorithm requires less than 1 s for cloud size lower than 20,000 points, hence, it is feasible for practical applications. Moreover, computational performance could be greatly improved on robots using high performing processors.

## VI. Experimental results

In this section, the use of the UPD for terrain traversability is demonstrated using different data sets acquired in indoor and outdoor environments. The test instrumentation includes an ASUS-EeePC-1215N featuring the Intel Atom D525 dual core processor, a NVIDIA ION discrete graphics processor and 3GB of RAM memory. In order to demonstrate its versatility and general applicability, it was applied to 3D point clouds generated by two sensors that differ in resolution, accuracy and field of view. Specifically, a Microsoft Kinect RGB-D camera was used for indoor acquisitions and integrated with the rover Dune, built at the Applied Mechanics Laboratory of the University of Salento, whereas the Point Grey XB3 stereo system was employed during outdoor testing mounted on an experimental tractor. The latter part of the research was performed within the European project Ambient Awareness for Autonomous Agricultural Vehicles (QUAD-AV), which aims to develop a driverless autonomous tractor (Milella *et al.*, 2013). While the Kinect camera has a depth camera with a resolution of  $640 \times 480$  and automatically provides 3D point clouds data using the OpenNI C++ library, the XB3 trinocular camera needs a pre-elaboration for generating point clouds from each stereo pair, including the following steps:

- *Rectification.* Each image plane is transformed so that pairs of conjugate epipolar lines become collinear and parallel to one of the image axes. Using rectified images, the problem of computing correspondences is reduced from a 2D to a 1D search problem, typically along the horizontal raster lines of

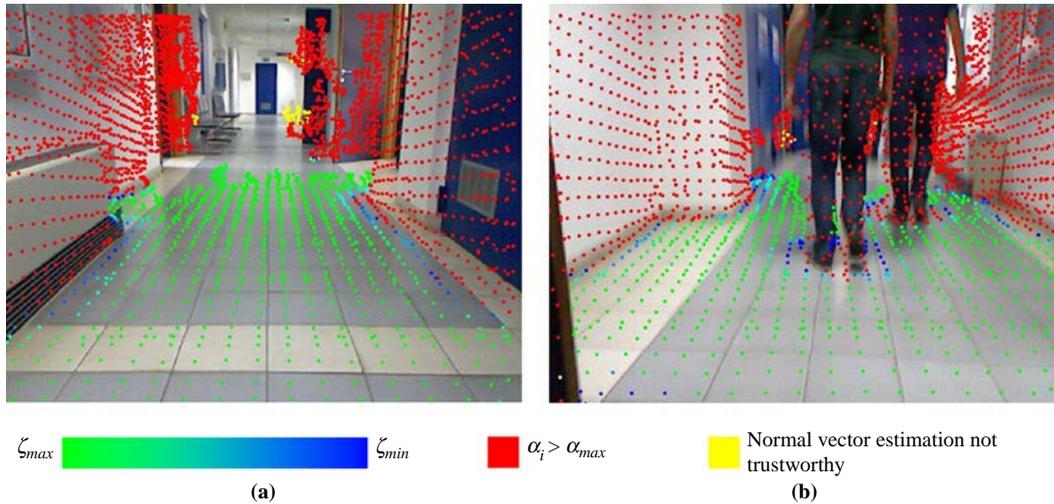
the rectified images. Rectification matrices are computed in the calibration step.

- *Disparity map computation.* To compute the disparity map a stereo block matching algorithm is used that finds corresponding points by a sliding sum of absolute difference (SAD) window.
- *3D point cloud generation in the reference camera frame.* Being the stereo pair calibrated both intrinsically and extrinsically, disparity values can be converted in depth values and 3D coordinates can be computed in the reference camera frame for all matched points.
- *Transformation from the reference camera frame to the vehicle reference frame.* In this stage 3D points are transformed from the camera frame to the vehicle frame.
- *Statistical filtering.* A statistical filter is applied to reduce noise and remove outlying points.
- *Voxelization.* In order to decrease the computational burden the number of points is reduced using a voxelized grid approach. A 3D voxel grid with a leaf size of 10 cm is created over the input point cloud space. Then, all the points in each voxel are approximated with their centroid.

### A. Indoor scenario

The indoor dataset corresponds to a survey in the hallway of the Department of Engineering for Innovation of the University of Salento. Figure 8 shows the result of the UPD approach for an indoor dataset, composed of 200 frames acquired by a Kinect camera at 1 Hz frequency, each RGB-D image is composed of approximately 300,000 points. In order to decrease the computational burden, the RGB-D image is filtered and downsampled. The algorithm runs on a filtered cloud composed of about 20,000 points. In these conditions, the ASUS-EeePC-1215N is able to elaborate about two frames per second.

In Figure 8, the original visual image acquired by the sensor is shown overlaid with the results obtained from the UPD-based analysis. Traversable areas are correctly denoted by green dots, geometric discontinuities are also flagged appropriately at the floor-wall intersections, using a color scale proportional to the estimated unevenness index as expressed by equation (7). As it can be seen from these figures, the UPD-based analysis

**Figure 8** Results of the UPD-based traversability evaluation

**Notes:** (a) Corridor scenario; (b) dynamic object detection; the classification results are expressed in color scale shown at the bottom of the figure; green dots denote traversable regions, red points denote non-traversable areas according to orientation analysis and yellow points mark unreliable observations

correctly detects the smooth floor and the static (walls and doors) and dynamic obstacles (humans) present in the scene.

In Table II, a quantitative evaluation of the classification results in terms of traversable or non-traversable points can be seen. The error rate was addressed through comparison with ground-truth data obtained by manual labeling. A subset of ten frames extracted from the indoor dataset was analyzed taking into account also Figure 8(a) and (b). The “ground detection error” is the percentage error between the points manually labeled as ground and the points interpreted as ground by the algorithm. The “non-ground detection error” is the percentage error between the points manually labeled as non-ground and the points marked as non-ground by the algorithm. Lastly, the “unreliable estimation” column refers to points in which the neighborhood does not contain enough information for the correct estimation of the UPD descriptor, according to the analysis proposed in Section IV.A. For indoor environments the percentage of unreliable points is low thanks to the high depth accuracy of the Kinect sensor. The error computed for the scenario of Figure 8(a) is relatively small due to the structured nature of the environment, 3.36 and 1.49 percent in ground and non-ground detection, respectively. While the error computed for the scene of Figure 8(b) is 7.52 and 6.85 percent in ground and non-ground detection, due to the presence of obstacles. However, when considering the average error computed over ten frames, the error results in 3.58 and 3.66 percent, respectively, for ground and non-ground detection.

**Table II** Detection error in indoor scenarios

Frame	Ground detection error (%)	Non-ground detection error (%)	Unreliable estimation (%)
Figure 8(a)	3.36	1.49	1.75
Figure 8(b)	7.52	6.85	0.53
Average (10 frames)	3.58	3.66	1.56

## B. Outdoor scenarios

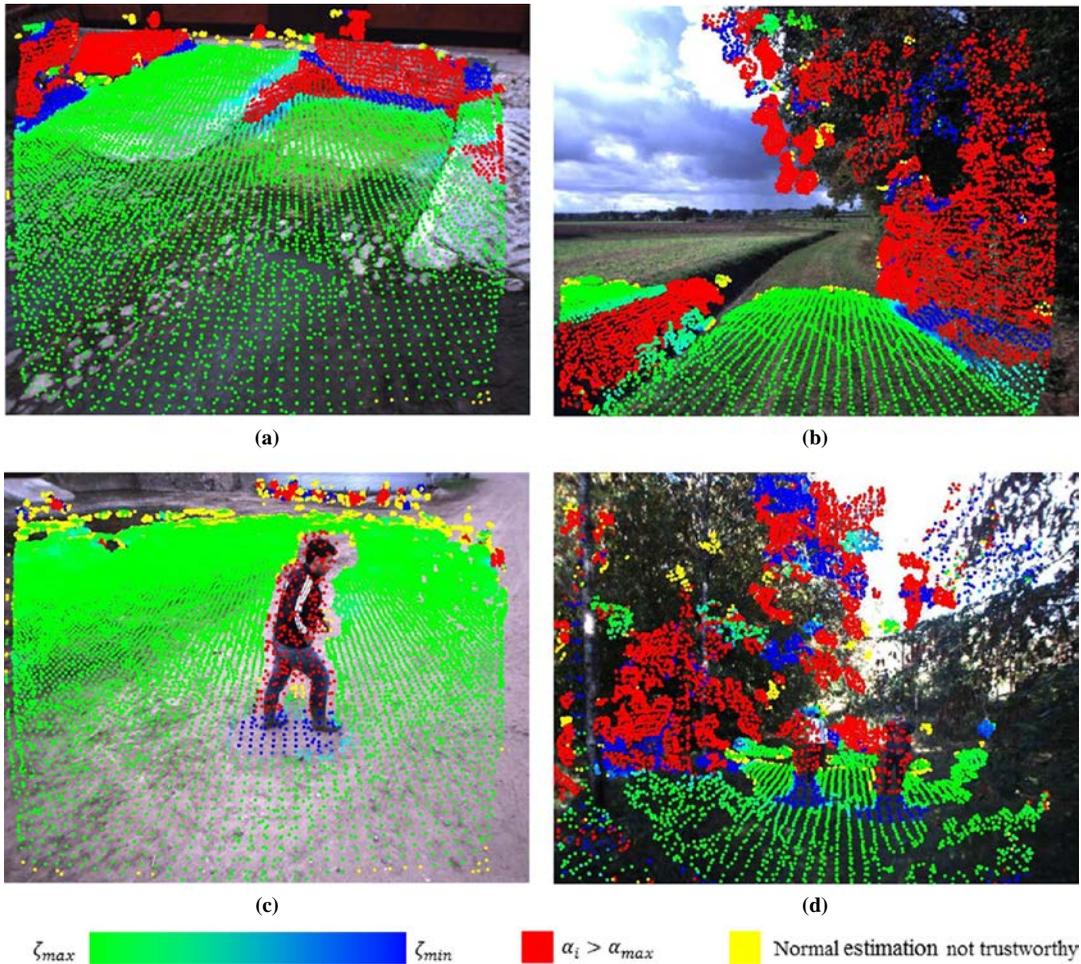
A dataset was acquired during a survey of typical agricultural environments using an experimental tractor, and it comprises of more than 1,000 point clouds generated using the XB3 stereo camera with a sampling rate of 2 Hz. The XB3 camera was mounted at the front of an experimental tractor. Due to the large amount of data, only some example experiments can be shown. Outdoor data refers to a larger field of view, and they are generally wider, less accurate and including a large number of outlier than indoor data. The following scenarios were encountered during the experiments:

- positive;
- negative; and
- dynamic obstacles.

In particular, a positive obstacle can be seen as an object higher than ground level, whereas a negative obstacle is lower than ground level. In the case of a positive obstacle the robot risks a collision; conversely a negative obstacle can be the cause of a crash. A dynamic object is something that suddenly appears in the scene interpretable as an obstacle, generally represented by human operators, other moving machineries or even animals. In this particular case the robot should be able to preserve safety.

Figure 9 shows the results obtained by the system in typical agricultural scenarios. The original image is overlaid with the results of the UPD analysis. Green dots mark traversable region. The color scale denotes the characteristics of traversability of the terrain according to the estimated unevenness index as expressed by equation (7). Black dots denote a region where the estimation of UPD is considered as not reliable as expressed by equation (10).

Let us refer to the scenario of Figure 9(a) including a relatively flat terrain with a low wall and a low-steep ramp. The wall is correctly flagged as a positive obstacle. It is worthwhile noting that the application of the conventional roughness index, i.e. the variance of the elevation value (Rohmer *et al.*, 2010), to this specific scenario would provide poor classification results

**Figure 9** Results of the UPD-based traversability evaluatio

**Notes:** (a) A low-steep ramp is classified as traversable by the system; (b) the ditch on the left is correctly interpreted as negative obstacle while the high vegetation on the right is interpreted as positive obstacle; (c) a walking human operator crosses the scene representing a dynamic obstacle; (d) complex scenario including trees and human operators; the classification results are expressed in color scale shown at the bottom of the figure: green dots denote traversable regions, red points denote non-traversable areas according to orientation analysis and yellow points mark unreliable observations

across the ramp that would be recognized as an obstacle. In contrast, the UPD-based approach correctly interprets the ramp as a traversable surface, due to the local smoothness and low global inclination. As already shown in the simulations discussed in Section V.A, the reason for the improvement in performance lies in the analysis of the local characteristics of the surface instead of the consideration of elevation values only. This makes the UPD more stable and robust than the standard roughness index. Figure 9(b) shows a scenario where the tractor is driving on a trail with a ditch on the left and high vegetation on the right. The UPD-based analysis correctly classifies as traversable the free path, whereas the negative and positive obstacles are marked as not traversable.

A different scenario is depicted in Figure 9(c), comprising relatively flat terrain and a walking human operator in front of the camera representing a dynamic object. The UPD-based classifier again provides consistent results: the terrain and the obstacles are correctly interpreted as traversable and non-traversable, respectively. The last case is shown in Figure 9(d), where two

human operators are surrounded by high vegetation. Even in this case, the UPD is able to correctly distinguish obstacles and free path.

Table III presents the quantitative evaluation of the algorithm for the outdoor dataset through comparison with manually labeled ground-truth data. The error is generally higher than the indoor dataset due to the lower depth accuracy of the XB3

**Table III** Detection error in outdoor scenarios

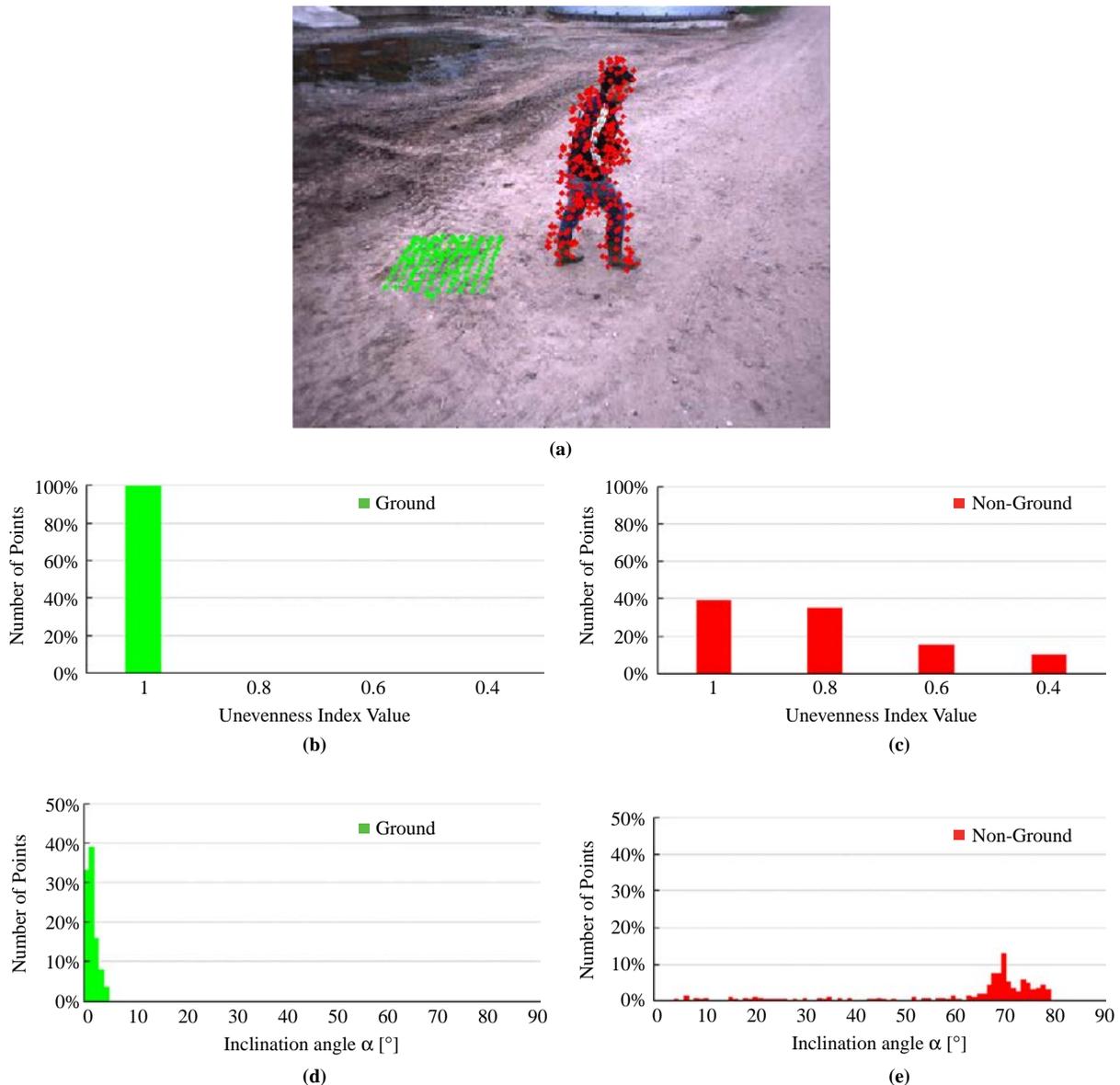
Figures	Ground detection error (%)	Non-ground detection error (%)	Unreliable estimation (%)
Figure 9(a)	4.68	7.67	2.99
Figure 9(b)	6.96	9.70	2.74
Figure 9(c)	2.49	13.61	11.13
Figure 9(d)	29.21	34.53	5.32
Average (10 frames)	10.42	15.97	5.54

camera. In detail, the error is relatively low for the scenario of Figure 9(a), in which the ground detection error is less than 5 percent. The error computed for the scene of Figure 9(b) increases due to the presence of the ditch on the left and the high vegetation on the right, yet it is less than 7 percent and still acceptable. In the scenario of Figure 9(c) the error of the non-ground detections increases, possibly due to the presence of high distance data where the accuracy of the stereovision drops. On the other hand, the ground detection error for the same scene is rather low, due to the presence of even terrain at short distance. A complex scenario is in Figure 9(d), in which, even though a large part of the ground was correctly detected, the error grows to around 30 percent due to the presence of a large

number of objects. Moreover, the scene presents numerous outlying points that may be a cause of misinterpretation. When considering the average error estimated over a selected subset of ten scenes, the results obtained by the UPD-classifier are robust showing that the algorithm can be used for outdoor scene interpretation.

A further evaluation of the performance of the algorithm is shown in Figure 10, in which two regions of the same image were compared through a histogram analysis showing the distribution of the features of the UPD. From the scenario shown in Figure 9(c), two 3D regions were manually extracted. The first region, denoted by green dots, refers to a flat patch of terrain, whereas the second region, denoted by red dots,

**Figure 10** Histogram of the UPD for two regions of the picture in Figure 9(b)



**Notes:** (a) The analyzed regions are projected over the camera image and marked as green and red dots denoting ground region and non-ground region, respectively; (b) histogram of the unevenness index values for the ground region; (c) histogram of the unevenness index for the non-ground region; (d) histogram of the inclination angle for the ground region; (e) histogram of the inclination angle for the non-ground region

corresponds to the 3D reconstruction of the operator in the scene, see Figure 10(a). Two histograms for each region were obtained analyzing the unevenness index, equation (7), and the inclination angle  $\alpha_q$  between the  $r$  vector and the  $z$ -axis, equation (11). The distribution of the unevenness index is plotted in Figure 10(b) and (c), for the ground and non-ground regions, respectively, whereas Figure 10(d) and (e) show the histogram of the inclination angle. The histograms show the number of analyzed points in percentage against the unevenness index and the inclination angle, respectively. Looking at the green bins of Figure 10(b), one can note that, the 100 percent of points are located close to the value 1, and this suggests high regularity of the region (ground). Conversely, red bins, in Figure 10(c), correspond to the region including the human operator, and its histogram exhibits a large spread across the  $x$ -axis, since it includes points having different characteristic of regularity. The result is also confirmed for the inclination angle histograms, in which the ground region is characterized by low angles, Figure 10(d), while the non-ground region is characterized by high inclination angles, Figure 10(e). As a matter of fact, Figure 10(d) shows that 100 percent of points feature an inclination angle less than  $5^\circ$ , whereas the bins in Figure 10(e) are concentrated around  $70$ – $80^\circ$  of inclination. The presence of a small concentration of bins scattered along all  $x$ -axis is justified by the different location of the points according to the shape of the obstacle.

## VII. Conclusion

The use of visual sensors and computer vision algorithms in the field of mobile robotics has been increasingly adopted for scene interpretation. Although a rich literature exists, more research is still required. In this paper, a new local descriptor has been proposed for scene interpretation based on the analysis of the distribution of the normal vector direction. The effectiveness of the UPD has been demonstrated in field experiments for terrain traversability assessment showing its advantages. The results obtained applying the proposed approach to indoor data indicates that the system produces highly reliable results on dense and clean data, but current stereo techniques cannot yet produce point clouds of comparable density and quality. For this reason, the results on outdoor environments were less accurate. The overall performance of the UPD showed that it could be used to enhance the ability of a robot to interpret images for navigation and obstacle negotiation purposes, since it may be easily integrated in a cost function for path planning purposes.

## References

- Bellone, M., Messina, A. and Reina, G. (2013), "A new approach for terrain analysis in mobile robot application", *IEEE International Conference on Mechatronics, Vicenza, Italy*, pp. 225–230.
- Braun, T., Bitsch, H. and Berns, K. (2008), "Visual terrain traversability estimation using a combined slope/elevation model", *IEEE/RSJ International Conference on Intelligent Robots and System, Berlin, Germany*, pp. 177–184.
- Dongshin, K., Sang, M.O. and James, M.R. (2007), "Traversability classification for UGV navigation: a comparison of patch and superpixel representations", *International Conference on Intelligent Robots and Systems, San Diego, CA, USA*, pp. 3166–3173.
- DuPont, E.M., Moore, C.A. and Roberts, R.G. (2008), "Terrain classification for mobile robots traveling at various speeds: an eigenspace manifold approach", *International Conference on Robotics and Automation, Pasadena, CA, USA*, pp. 3284–3289.
- Halatci, I., Brooks, C.A. and Iagnemma, K. (2007), "Terrain classification and classifier fusion for planetary exploration rovers", paper presented at IEEE Aerospace Conference, Big Sky, MT, USA.
- Howard, A. and Saraji, H. (2001), "Vision-based terrain characterization and traversability assessment", *Journal of Robotic System*, Vol. 18 No. 10, pp. 77–587.
- Howard, A., Turmon, M., Matthies, L., Tang, B. and Angelova, A. (2007), "Towards learned traversability for robot navigation: from underfoot to the far field", *Journal of Field Robotics*, Vol. 23 Nos 11/12, pp. 1005–1017.
- Kim, J., Lee, J. and Lee, J. (2010), "Terrain classification for improving traversability of autonomous vehicles", paper presented at International Conference on Robotics and Automation, Anchorage, AL, USA.
- Kinect (2010), Microsoft Kinect, available at: [www.xbox.com/en-US/kinect](http://www.xbox.com/en-US/kinect)
- Kriegel, H.P., Kröger, P., Schubert, E. and Zimek, A. (2008), "A general framework for increasing the robustness of PCA-based correlation clustering algorithms", paper presented at 20th Int. Conf. on Scientific and Statistical Database Management (SSDBM), Hong Kong, China.
- Larson, J., Trivedi, M. and Bruch, M. (2011), "Off-road terrain traversability analysis and hazard avoidance for UGVs", paper presented at IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany.
- Milella, A., Reina, G. and Foglia, M.M. (2013), "A multi-baseline stereo system for scene segmentation in natural environments", in *Proceedings of IEEE International Conference on Technologies for Practical Robot Applications (TEPRA)*.
- Milella, A., Reina, G. and Siegwart, R. (2006), "Computer vision methods for improved mobile robot state estimation in challenging terrains", *Journal of Multimedia*, Vol. 1 No. 7, pp. 49–61, November/December.
- Neuhaus, F., Dillenberger, D., Pellenz, J. and Paulus, D. (2009), "Terrain drivability analysis in 3D laser range data for autonomous robot navigation in unstructured environments", paper presented at IEEE Conference on Emergin Technologies & Factory Automation ETFA, Mallorca, Spain.
- Osari, H. (2003), "A new method for assessing land leveling to produce high quality consolidated paddy fields", *Paddy Water Environ*, Vol. 1 No. 1, pp. 35–41.
- Papadakis, P. (2013), "Terrain traversability analysis methods for unmanned ground vehicles: a survey", *Engineering Application of Artificial Intelligence*, Vol. 26, pp. 1373–1385.
- Pauly, M. and Kobbelt, L. (2002), "Efficient simplification of point-sampled surfaces", *Proceedings of the Conference on Visualization, Washington, DC, USA*.
- Reina, G. and Milella, A. (2012), "Towards autonomous agriculture: automatic ground detection using trinocular stereovision", *Sensors*, Vol. 12, September, pp. 12405–12423.
- Reina, G., Milella, A. and Underwood, J. (2012), "Self-learning classification of radar features for scene understanding", *Robotics and Autonomous System*, Vol. 60, pp. 1377–1388.
- Reina, G., Ishigami, G., Nagatoni, K. and Yoshida, K. (2010), "Odometry correction using visual slip-angle estimation for planetary exploration rovers", *Advanced Robotics*, Vol. 24, No. 3, pp. 359–385.

- Rohmer, E., Reina, G. and Yoshida, K. (2010), "Dynamic simulation-based action planner for a reconfigurable hybrid leg-wheel planetary exploration rover", *Advanced Robotics*, Vol. 24 Nos 8/9, pp. 1219-1238.
- Rusu, B.R. (2009), "Semantic 3D object maps for everyday manipulation in human living environments", dissertation, Technical University Munich.
- Rusu, B.R. and Cousins, S. (2011), "3D is here: point cloud library (PCL)", paper presented at IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China.
- Rusu, B.R., Brandiski, G., Thibaux, R. and Hsu, J. (2010), "Fast 3D recognition and pose using the viewpoint feature histogram", *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan.
- Seraji, H. (1999), "Traversability index: a new concept for planetary rovers", *IEEE Intern. Conf. on Robotics and Automation, Detroit, MI*, pp. 2006-2013.
- Silver, D., Sofman, B., Vandapel, N., Bagnell, J.A. and Stentz, A. (2006), "Experimental analysis of overhead data processing to support long range navigation", paper presented at International Conference on Intelligent Robots and System, Beijing, China.
- Vandapel, N., Huber, D.F., Kapuria, A. and Hebert, M. (2004), "Natural terrain classification using 3-D lidar data", *IEEE International Conference on Robotics and Automation, New Orleans, LA, USA*, pp. 5117-5122.

### About the authors

**Mauro Bellone** received the MS degree in automation engineering from the University of Salento, Lecce, Italy, in 2009, where he is currently working toward a PhD degree in mechanical and industrial engineering. In 2009, he worked at the Space Robotics Laboratory of Tohoku University, Sendai, Japan during an internship financially supported by the University of Salento. His interests include mobile robotics, computer vision and control systems. His research concerns

the area of advanced sensorial perception for mobile robotics. Mauro Bellone is the corresponding author and can be contacted at: mauro.bellone@unisalento.it

**Giulio Reina** received the Laurea degree and the Research Doctorate degree from the Politecnico of Bari, Italy, in 2000 and 2004, respectively, both in mechanical engineering. From 2002 to 2003, he worked at the University of Michigan Mobile Robotics Laboratory as a Visiting Scholar. In 2007, he was awarded a Japanese Society for Promotion of Science fellowship for a 1-year research at the Space Robotics Laboratory of the Tohoku University, Sendai, Japan. In 2010 Dr Reina was also selected to receive an Endeavour Research Fellowship at the Australian Centre for Field Robotics of the University of Sydney, Australia. Currently, he is an Assistant Professor in Applied Mechanics with the Department of Engineering for Innovation of the University of Salento, Lecce, Italy. His research interests include mobile robotics for planetary exploration, mobility and localization on rough-terrain, advanced perception systems, and agricultural robotics.

**Nicola Ivan Giannoccaro** received his MS degree in electronic engineering and his PhD degree in advanced production systems from the Politecnico of Bari, Bari, Italy, in 1996 and 2000, respectively. Since 2001, he has been a Lecturer with the Department of Innovation Engineering, University of Salento, Lecce, Italy. His research interests include mechatronics systems, control of mechanical systems, modal analysis and dynamical identification.

**Luigi Spedicato** received his MS degree in automation engineering from the University of Salento, Lecce, Italy, in 2009, and his PhD in mechanical and industrial engineering in 2013. His interests include sensing, signal processing and control systems. His research pertains to mechatronics and mobile robotics.