Mind the gap: detection and traversability analysis of terrain gaps using LIDAR for safe robot navigation

Arnab Sinha, Panagiotis Papadakis

Mind the gap: detection and traversability analysis of terrain gaps using LIDAR for safe robot navigation

Arnab Sinha and Panagiotis Papadakis*

ALCOR, Vision, Perception and Cognitive Robotics Laboratory, Department of Computer, Control and Management Engineering, University of Rome, "La Sapienza," Italy

SUMMARY

Safe navigation of robotic vehicles is considered as a key pre-requisite of successful mission operations within highly adverse and unconstrained environments. While there has been extensive research in the perception of positive obstacles, little progress can be accredited to the field of negative obstacles. This paper hypostatizes an elaborative attempt to address the problem of negative obstacle detection and traversability analysis in the form of gaps by processing 3-dimensional range data. The domain of application concerns Urban Search and Rescue scenarios that reflect environments of increased complexity in terms of diverse terrain irregularities. To allow real-time performance and, in turn, timely prevention of unrecoverable robotic states, the proposed approach is based on the application of efficient image morphological operations for noise reduction and border following the detection and grouping of gaps. Furthermore, we reason about gap traversability, a concept that is novel within the field. Traversability assessments are based on features extracted through Principal Component Analysis by exploring the spatial distribution of the interior of the individual gaps or the orientation distribution of the corresponding contour. The proposed approach is evaluated within a realistic scenario of a tunnel car accident site and a challenging outdoor scenario. Using a contemporary Search and Rescue robot, we have performed extensive experiments under various parameter settings that allowed the robot to always detect the real gaps, and either optimally cross over those that were traversable or otherwise avoid them.

KEYWORDS: Mobile robots; Navigation; Motion planning; Computer vision; Man–machine systems.

1. Introduction

In parallel to common applications where mobile robots operate in indoor-structured environments, there has been an evident interest in advancing robot technology to increase the degrees of freedom in their operation so that they can be deployed within outdoor, off-road, natural as well as unnatural environments. Typical scenarios concern planetary exploration, military, forestry, agriculture, and mining, together with Urban Search and Rescue (USAR) robotics, which is the focus of this paper.

Robots are able to operate in challenging environments by using reconfigurable components that (passively or actively) adapt to rough terrain. In such applications, several issues need to be addressed such as: (i) assessment of terrain traversability, (ii) planning optimal paths with respect to given criteria and (iii) automatically adapting the articulating parts of the robot.

The aim of the Natural Human-Robot Cooperation in Dynamic Environments (NIFTi) project, where the present work contributes to, is to develop a robotic system that teams with human operators and firefighters for first-response missions in USAR. Our first extensive experience with the consortium's robotic platform[2] was at a joint exercise in July 2011[15] and recently within an end-user evaluation taking place in a tunnel at the VVF training site in Montelibretti in December 2011[9] (see Fig. 1). The scenario spanned a broad area into the tunnel filled with debris, pallets, barrels, crashed vehicles and smoke. The overall setting comprised various hazards for the mobility of the robot, among which were several gaps, some of which were traversable and others were not.

A collective observation was that due to the low point of view from on-board sensors, constrained lighting and presence of smoke, it was very difficult for the users to manually navigate the robot, let alone perceive and avoid or traverse gaps in an optimal way. This strived the need to improve the autonomous navigation capabilities of the robot, but in a way that it would be transparent to the user so that it could be trusted.

In this work we address the problem of *negative obstacle perception* and *traversability analysis* in the form of gaps, hence dealing with the first highlighted issue, namely, assessing terrain traversability. This is certainly one of the most challenging perception problems, since the presence of a negative obstacle can only be inferred through the absence of data that can have various interpretations.

We propose a methodology that is applicable in real-time and provides accurate traversability assessments under various challenging conditions. Our approach for gap perception and analysis is based on 3-dimensional (3D) point cloud processing, which is considered to be more reliable in general compared to other methods that are based either on vision or other sensor modalities. From this perspective and in comparison to previous methods that rely on 3D scene

* Corresponding author. E-mail: papadakis@dis.uniroma1.it

Fig. 1. (Colour online) Tunnel car accident at Montelibretti's firefighter Italian school used for USAR evaluation.

analysis, the main contributions of the present work are summarized as follows:

- Organization/grouping of regions that correspond to gaps.
- Traversability analysis of gaps allowing for non-binary and higher level classification.
- Extensive experiments in USAR scenarios.

In detail, we employ 2D image morphological and contour detection (through border following) algorithms for perceiving gaps spread around the ground in the vicinity of the robot, without the need of processing the complete 3D scene, unless gap detection is required for more than one plane. For each detected gap that is captured by the corresponding contour, we perform a traversability analysis based on its eigen-decomposition, where we consider the spatial distribution of the interior of individual gaps or the orientation distribution of the corresponding contour to assess the optimal traversal path accounting for vehicle mobility constraints.

The remainder of the paper is organized as follows: In Section 2 we first review the previous works in negative obstacle detection and in particular for gap perception to motivate the directions that we followed in the proposed methodology. In Section 3 we proceed to the detailed description of the proposed gap perception and traversability analysis approach, and finally in Section 4 we present the results of an extensive set of experiments within a tunnel car accident scenario and an outdoor environment, and elaborate on the performance of our approach.

2. Previous Works

In addressing the gap detection problem, various directions have been explored, nonetheless the problem remains challenging, especially for Search and Rescue environments that are highly diverse and unconstrained in terms of terrain irregularities. With respect to the sensor modalities that have been employed for sensing the environment, we may tabulate the previous works as shown in Table I.

Table I. Sensors used in earlier approaches.

Earlier work	Stereo camera	Laser	Other sensors
Matthies *et al.*[13]	Yes	No	No
Bellutta *et al.*[1]	Yes	No	No
Matthies *et al.*[14]	Yes	No	Thermal
Dima *et al.*[5]	Yes	Yes	Infrared camera
Kelly *et al.*[8]	Two	Four	Omnidirectional camera, monochrome digital camera
Crane *et al.*[4]	**No**	**Yes**	**No**
Dubbelmanand *et al.*[6]	Yes	No	No
Heckman *et al.*[7]	**No**	**Yes**	**No**
Larson *et al.*[11,12]	**No**	**Yes**	**No**

Probably the first elaborative attempt to deal with this problem can be attributed to the work of Matthies *et al.*,[13] where the presence of negative obstacles is inferred by performing ray tracing for every pixel within the range image, and comparing the actual range values along the ray with the expected range values according to the position of the ground plane. If the distance between the ranges of consecutive pixels was greater than a threshold, then this indicated the presence of negative slope or ravine.

The approach of Bellutta *et al.*[1] for terrain perception was based on the combination of geometric and visual features through a rule-base system. Terrain was geometrically classified into negative or positive obstacles by inspection of the height profile of elevation data, while the terrain support was statistically learned through expectation maximization in colour space.

An alternative approach in terms of perception for detection of negative obstacles during night was later proposed in ref. [14], wherein range data were combined with thermal features of the terrain that highlight cavities as potential negative obstacles. The method was based on the observation that negative obstacles retain more heat during night than planar surfaces.

Dima *et al.*[5] used feature and classifier fusion for obstacle detection and terrain traversability where the basis features that are computed for various perceptual modalities correspond to the mean and variance of pixel values along a set of image patches that span the whole image. Combining features that incorporate domain knowledge,[10] different classifier fusion strategies are evaluated that show improved classification scores for road, human, and negative obstacle detection in comparison with single feature-based classifiers.

Kelly *et al.*[8] describe the design and operation of a human–robot team for off-road navigation, wherein terrain classification is based on geometry-based features combined with multi-spectral image-based features. The robot-support surface is extracted by ray-tracing of laser-beams and training a neural network to derive the load-bearing surface when traversing over vegetated areas, while negative obstacles

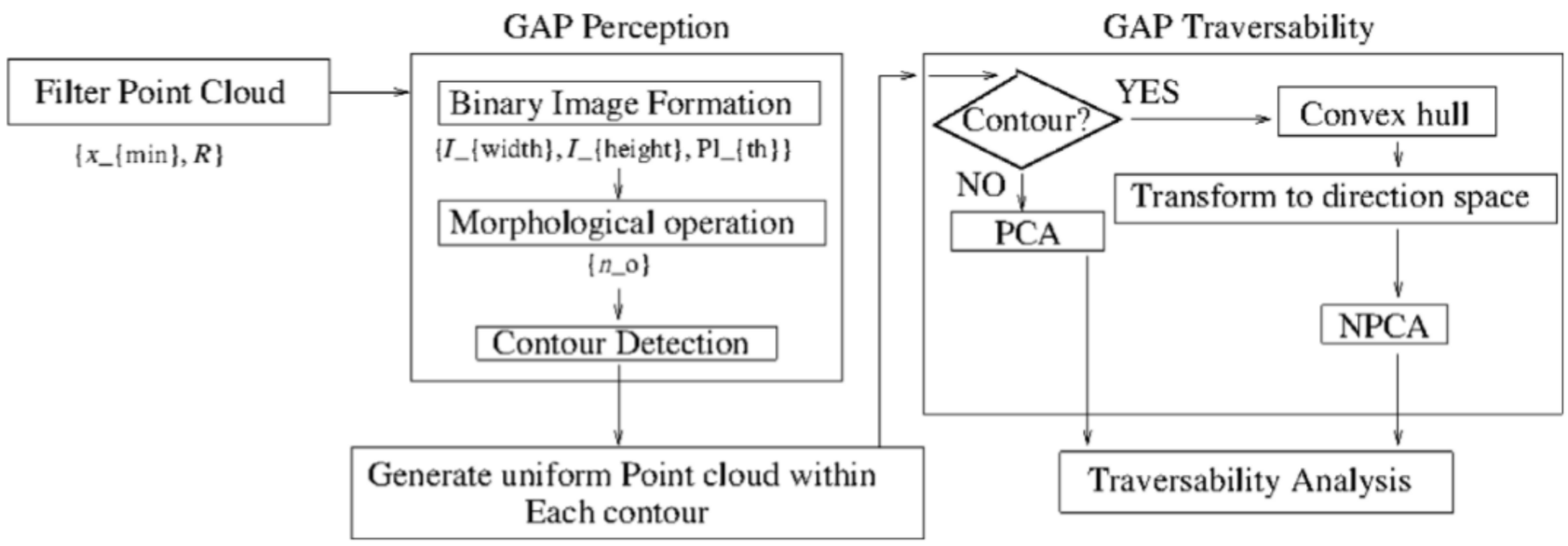


Fig. 2. The modules of the proposed methodology. The Gap perception block perceives the gaps with the parameters written below each sub-block traversability analysis can be performed in two ways namely, with contour information or using the space within the gap.

are found by the absence of laser hits in the direction perpendicular to the support surface.

Crane *et al.*[4] describe the *traversability grid* data structure that served as a common interface between components that either produced or consumed perception data. A collection of smart sensors complied with a fixed underlying traversability assessment protocol which assigned traversability scores to each grid cell. Three LIDAR sensors were employed together with a camera sensor, with distinct traversability analysis roles. A LIDAR sensor directed toward the ground was dedicated for detecting negative obstacles based on the assumption that the unmanned ground vehicle (UGV) follows a level path, hence cells lying below the prescribed plane were assigned traversability values mainly based on their range distance.

In ref. [6], obstacles were detected using dense 3D terrain data reconstructed from stereo disparities in the direction of image columns. First, a disparity validity measure was employed together with an image pyramid to produce reliable disparity estimates. In the following, the traversability was computed for each pixel of the disparity image by estimating the maximum vertical (positive or negative) slope and using hysteresis thresholding that was driven by morphological opening and region filling.

Heckman *et al.*[7] performed detection of potential negative obstacles by initially performing ray-tracing for occlusion labeling and finally for context-based labeling. Given a 3D voxel grid where cells were classified into *linear*, *surface* and *scatter*, ray-tracing was used to propagate the class of occupied voxels to the corresponding occluded voxels whereas context-based labeling was used to differentiate between four cases that could be the cause of data absence and hence reason about the presence of negative obstacles.

In the work of Larson *et al.*,[12] terrain traversability was determined by the presence of positive–negative obstacles, step edge obstacles, slope steepness and terrain roughness. Patches of missing range data that exceeded some size were considered as potential negative obstacles, and a consecutive filtering process determined whether these could be the result of *shadowing* from positive obstacles. Larson and Trivedi[11] in their work explored a two-stage (long and short-range) negative obstacle detection framework. Initially, potential negative obstacles were detected at a distance using the NODR classification approach and then further refined and filtered using support vector machines (SVM) when the UGV has sufficiently approached the surrounding area. Negative Obstacle DetectoR (NODR) comprises a multi-pass detection process that first looked for steps and next for gaps whose characteristics could either be directly measured from the available range data, or inferred by using contextual cues, such as sudden negative or positive elevation drops. Eventually, using an SVM model trained on ground truth data, true and false positives of negative obstacles were distinguished once the UGV had sufficiently approached.

As can be derived from Table I, there are only three works that correspond to same sensor allotment (highlighted in bold). The basis of the corresponding works as described in refs. [4, 7, 11, 12] corresponds to laser ray tracing. Unfortunately, for lack of a generative framework and due to their dependence on the underlying laser device, a straightforward comparison seems infeasible. In contrast to earlier works, we propose a gap detection method that can be used by any laser device. Moreover, all previous works discussed so far concern applications where robotic vehicles operate within natural environments. To the authors' knowledge, there is no previous work on gap perception regarding USAR scenarios wherein the complexity of the terrain is by far more increased, and no previous work in gap traversability analysis regardless of the environment of operation. From this perspective, the present work constitutes an important step toward a better comprehension of the problem and proposes a solution that is robust under diverse conditions.

3. Proposed Gap Detection and Traversability Analysis Methodology

The problem that we are addressing is decomposed into two sub-problems, namely, *gap detection* and *traversability analysis*. In Fig. 2, we provide a schematic overview of individual steps followed in the proposed methodology.

Gap detection: Given a 3D point cloud $P = \{\mathbf{p}_i | \mathbf{p}_i = (x_i, y_i, z_i), i = 1, 2, \ldots, N_P\}$, we seek to detect sets of point clouds G_j, $j = 1, 2, \ldots, N_g$ that correspond to the gaps in the vicinity of the robot, where N_P denotes the total number of points, and N_g is the total number of different gaps detected.

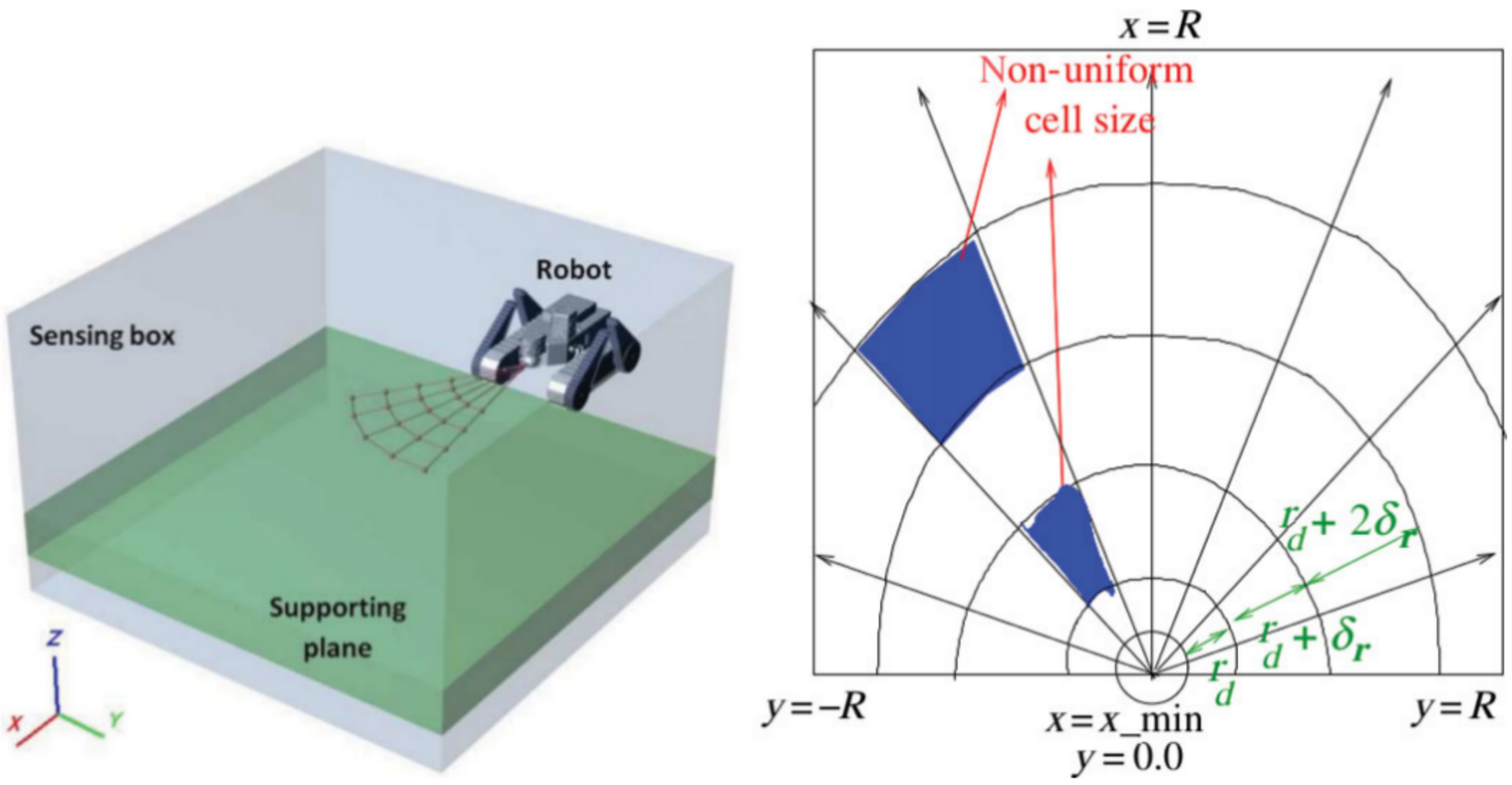


Fig. 3. (Colour online) Constrained sensory box in front of a robot.

Gap traversability: We assess the traversability–mobility of the gaps, that is, we reason about whether it is safe for a robot to traverse a detected gap as well as derive the optimal traversal mode. This information is stored for each gap within a vector $\mathbf{t}_j = \{t, \mathbf{p}_{\text{center}}, \mathbf{p}_{\text{start}}, \mathbf{p}_{\text{end}}\}$, where $t \in \{0, 1\}$ designates whether a gap is traversable or not, and in the former case, $\mathbf{p}_{\text{center}}$ provides the center of the gap contour together with the optimal START and END poses, $\mathbf{p}_{\text{start}}$ and $\mathbf{p}_{\text{end}}$, respectively, that the robot should follow to traverse over the gap.

An implicit assumption that we make here, in accordance to ref [7] which shares most familiarity to the present work, is that the set of points belonging to a given gap contour satisfy a plane equation up to a certain error constant in terms of their distance from that plane. In other words, we restrict the problem of gap detection from the complete 3D scene to a ground slice of fixed thickness in front of the robot, which is eventually represented by the projections of 3D points within the slice onto the 2D ground plane equation.

3.1. Gap detection

The gap detection stage has been split into the following three parts to assist in the comprehension of overall methodology:

- Binary image formation (Section 3.1.1)
- Image processing (Section 3.1.2)
- Gap point cloud generation for further processing (Section 3.1.3)

The pseudo-code of the overall methodology is given in Algorithm 1, and finally in Section 3.1.4 we describe the computational complexity of the proposed approach.

3.1.1. Binary image formation. We begin by constraining the space of gap detection from the complete $\Re^3$ to the space within a virtual sensory box defined as $[x_{\min}, R] \times [-R, R] \times [-z_{\min}, z_{\max}] \subset \Re^3$ (as shown in Fig. 3 (left)) according to the specifications and dimensions of robotic platform used in our experiments[2] (we elaborate more on this in Section 4.2). In general, $z_{\min}$ can be thought of as infinity, or the maximum range distance that a particular laser sensor can support.

Our gap detection approach is applied onto the set of points that reside close to a 2D plane that could support the robotic vehicle. We detect these points and project them onto a 2D binary image I that will be used in next steps (Sections 3.1.2 and 3.1.3).

One could attempt to estimate the robot supporting plane by fitting a plane into the point cloud within the sensory box, using, for example, regression, maximum likelihood or RANSAC. However, it is inherently implausible to follow this approach. Due to the presence of gaps, the supporting plane would not be easily distinguishable due to lack of sufficient data (inliers) that could result in an erroneous plane estimation. Instead, a more suitable approach could be to consider a fixed planar area in front of the robot. In detail, we take a slice of terrain mainly for two reasons, namely, (i) to compensate for a small variance in the real 3D position of coplanar points that could be due to error or noise, and (ii) to allow the perception of gaps not only within perfectly planar terrain in front of the robot of zero inclination, but also within slightly inclined but highly planar terrains in the foreground. The plane that corresponds to this slice is estimated by taking into account the 3D pose estimation of the robot that is computed by fusing sensory information from an Inertia Measurement Unit (IMU) and the registration of 3D point clouds as acquired by the LIDAR sensor. A particular decision on how to regress the plane depends on the density of laser sensing, the dimensions of the sensing box and the nature of the terrain that is expected to be encountered. Without loss of generality, we continue the description of our approach on the basis of an underlying plane estimation process.

Using the standard notation $ax + by + cz + d = 0$ for the representation of a 3D plane, by employing a threshold Pl_{th} a point $\mathbf{p}_i$ can be checked whether it resides on the robot's supporting plane by using the following equation:

$$t_i = \frac{ax_i + by_i + cz_i + d}{\sqrt{a^2 + b^2 + c^2}}. \tag{1}$$

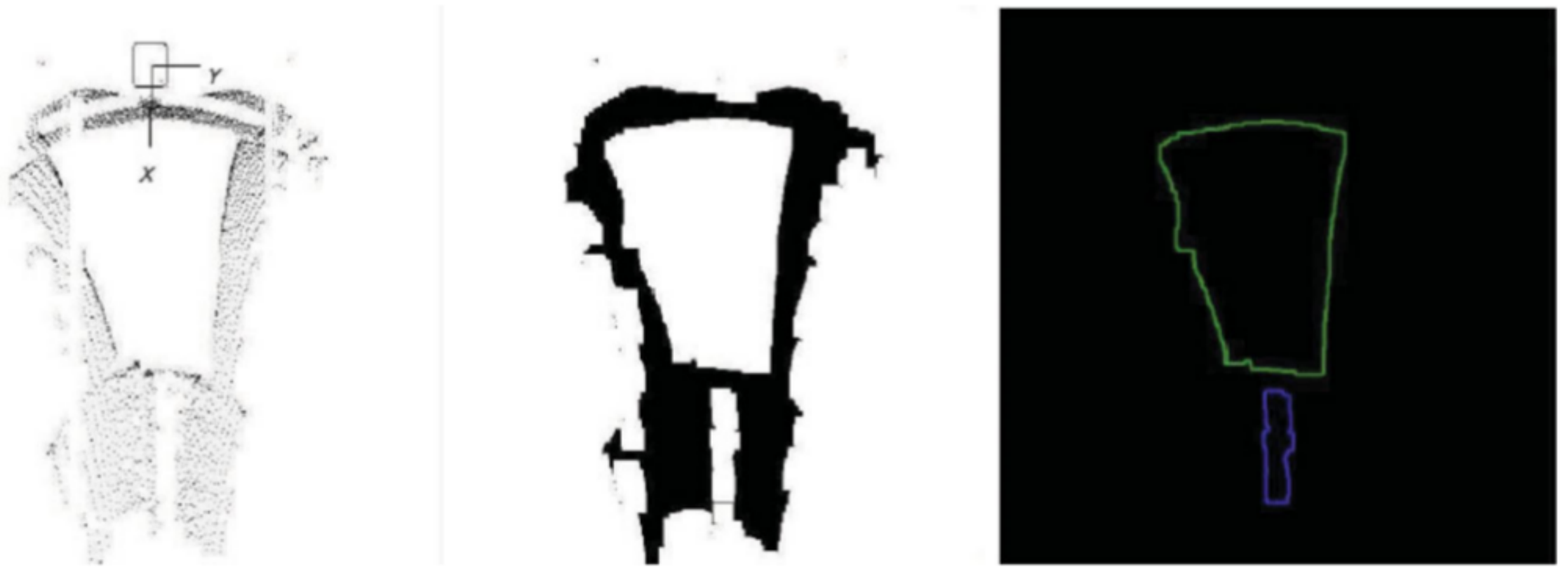


Fig. 4. (Colour online) Consecutive stages of the gap detection process. Left: Binary image of the plane point cloud within the polar coordinate space, the rectangle shows the robot position and the X, Y axes with respect to robot. Middle: After morphological operations. Right: Detected gap contours.

If $|t_i| \leq Pl_{\text{th}}$ then the ith point belongs to the robot supporting plane, otherwise if $t_i > Pl_{\text{th}}$, the point represents a positive obstacle. It could be argued that it is not appropriate to fix a threshold for positive obstacles or that a single point cannot represent an obstacle. This issue is alleviated as described in the next section, where by means of image processing we can filter out noisy points and outliers. Moreover, as shown in Algorithm 1, once $\mathbf{p}_i$ is classified as a positive obstacle point, we further mark those pixels of the binary image that correspond to the points behind $\mathbf{p}_i$ along the ray that connects the laser's origin and $\mathbf{p}_i$ as NON gap pixels in order to not misinterpret as gaps the absence of data that is due to positive obstacle occlusion.

Finally, given a point $\mathbf{p}_i$, the computation of its corresponding image pixel coordinates (k_i, l_i) is performed by the following:

$$k_i = \left\lfloor \frac{\tan^{-1}(\frac{y_i}{x_i})}{\theta_{\text{int}}} \right\rfloor, \tag{2}$$

$$l_i = \left\lfloor \log_{\delta_r}\left(\frac{r_i}{r_d} + 1\right) \right\rfloor, \tag{3}$$

$$r_i = \sqrt{{x_i}^2 + {y_i}^2}, \tag{4}$$

$$r_d = \frac{R}{\delta_r^{I_{\text{height}}-1} - 1}. \tag{5}$$

In the above equations, parameter $\theta_{\text{int}} = \pi / I_{\text{width}}$ denotes the step between two consecutive θ values as shown in Fig. 3 (right). Let us assume that the radial values are $r_1, r_2, \ldots, r_n$. If we set $r_2 - r_1 = r_d$ as the initial interval, the remaining consecutive intervals are $r_3 - r_2 = r_d + \delta_r$, $r_4 - r_3 = r_d + 2\delta_r$ and so on. Since the maximum radial value is fixed to R and also the resolution (number of radial values) is fixed by the parameter I_{height}, the two parameters r_d and δ_r are interdependent. We fix the parameter δ_r (discussed in Section 4.2) and evaluate r_d as shown in Eq. (5).

3.1.2. Image processing. We proceed by applying a noise filtering operation to the 2D binary image that was computed in the previous step. Due to the fact that this image is formed by employing a hard threshold Pl_{th}, the image will not be smooth in the sense that there may be some regions with points belonging to the robot supporting terrain but have been excluded. Moreover, there could be some regions where some spurious obstacle points have been detected and resulted in a noisy image. At this point, we are making a *local spatial coherence* assumption, that is, in the presence of a "gap" or an "obstacle," the corresponding feature should be present in the neighbourhood as well. Following this idea, we smooth the binary image with binary image morphological filters. First we apply an *erosion* filter and then a similar *dilation* filter. The filtered image can be termed as I_D.

Finally, we detect the gaps in the filtered image $I_D(i, j)$ by a state-of-the-art contour detection algorithm proposed by Suzuki *et al.*,[20] which is based on the idea of border following. This method provides an accurate estimate of outer contours and holes, and a very efficient implementation can be found within the OpenCV library.[3] In Fig. 4 we demonstrate the consecutive stages of gap contour extraction in a representative example.

3.1.3. Gap point cloud extraction. At the final step, we assemble the gap point clouds G_j that are going to be used as input in the "gap traversability analysis" stage. In that stage (detailed in Section 3.2), we may require the interior of the gaps or only the gap contour. In Algorithm 1, this condition is controlled by the boolean variable K. The conversion from an image pixel (i, j) to the corresponding 3D point coordinates (x, y, z) is easily obtained by inversion of Eqs. (2)–(4).

3.1.4. Computational complexity. The computational complexity of the binary image formation algorithm is linear to the total number of points N_P, hence $O(N_P)$. Erosion and dilation operations are performed on a binary image, hence the computational complexity is $O(I_{\text{width}} \times I_{\text{height}})$, where I_{width} and I_{height} correspond to the width and height dimensions of the image respectively. The complexity of the border following algorithm by Suzuki *et al.*[20] for contour detection is linear to the number as well as the length of contours. We may therefore implicitly disregard the latter cost as it is dominated by the computational complexity of binary image-formation which is linear to the total number of 3D points whose numbers greatly exceed the total number of points belonging to the contours. The total worst-case complexity of the proposed algorithm is $O(N_P + I_{\text{width}} \times I_{\text{height}})$. Since we eventually fix the resolution of the image, namely, I_{width} and I_{height}, the overall complexity of the

Algorithm 1: Gap point cloud extraction

Input:
$P = \{\mathbf{p}_i\}$: Point cloud
n_o: number of morphological operations
$\{a, b, c, d\}$: Robot supporting plane extraction
K: Boolean (TRUE: if interior, FALSE: if contour)
Output:
$G = \{G_i\}$: Set of point clouds corresponding to each gap G_i
begin
  Transform point cloud P to robot local coordinate system: $P --> Q$
  Filter point cloud Q according to sensory box: $Q --> S$
  $I = ones$: Initialize the binary image.
  for *Each point* $\mathbf{s}_i$ *within point cloud* S **do**
    if $\mathbf{s}_i$ *point is on robot support plane* **then**
      Find (θ_i, r_i) corresponding to $\mathbf{s}_i$
      $I(\theta_i, r_i) = 0$: Mark the pixel as "not a GAP"
    end
    else if *(*$\mathbf{s}_i$ *point is from a positive obstacle)* **then**
      Find (θ_i, r_i) corresponding to $\mathbf{s}_i$
      for *all pixels behind (occluded) positive obstacle:* $r = r_i$ *to* I_h **do**
        $I(\theta_i, r) = 0$: Mark the pixel as "not a GAP"
      end
    end
  end
  $I_D = \text{Morphological\_operation}(I, n_o)$
  $C = \{C_j; j = 1, \ldots, N_g\}$ FindContour(I_D)
  if *(K=TRUE)* **then**
    for *each contour* $C_j \in C$ **do**
      Generate **interior** point cloud G_j according to the interior of contour C_j
    end
  end
  else if *(K=FALSE)* **then**
    for *each contour* $C_j \in C$ **do**
      Generate **contour** point cloud G_j according to the contour C_j
    end
  end
end

Algorithm 2: Gap traversability analysis

Input: G_j: Point cloud representing the gap
Output: $\mathbf{t}_j$: Traversability analysis data for the gap
begin
  if *Contour* **then**
    $Con_j = \text{convex}(G_j)$: Extract convex point cloud
    $Ct_j = \text{Contour}(Con_j)$: Generate uniform contour point cloud
    $\mathbf{m}_j = \text{centroid}(Ct_j)$: Required for START and END pose evaluation
    $D_j = \text{direction}(Ct_j)$: Extract direction vectors
    $Cov_j = \text{covariance}(D_j)$
    $[\mathbf{e}, \lambda] = \text{PCA}(Cov_j)$: Eigen-vectors $\mathbf{e}$ and eigen-values λ of the gap contour orientation
    $\mathbf{e}_{\text{opt}} \leftarrow$ 1st eigen-vector
  end
  else if *Interior* **then**
    $\mathbf{m}_j = \text{centroid}(G_j)$
    $Cov_j = \text{covariance}(G_j - \mathbf{m}_j)$
    $[\mathbf{e}, \lambda] = \text{PCA}(G_j)$: Eigen-vector $\mathbf{e}$ and eigen-value λ of the gap
    $\mathbf{e}_{\text{opt}} \leftarrow$ 2nd eigen-vector
  end
  $G'_j = \{g'_{j,k} = \mathbf{g}_{j,k}\mathbf{e}^T_{\text{opt}}\}$: projected points onto the optimal principal direction $\mathbf{e}_{\text{opt}}$
  $len_j : \max_{k,l} ||g'_{j,k} - g'_{j,l}||$
  if $len_j > d_r$ **then**
    $\mathbf{t}_j(1) = 0$
  end
  else
    $\mathbf{t}_j(1) = 1, \mathbf{t}_j(2:4) = \mathbf{m}_j$
    $START = \mathbf{m}_j + \eta\mathbf{e}_{\text{opt}}$
    $END = \mathbf{m}_j - \eta\mathbf{e}_{\text{opt}}$
    if $||START|| > ||END||$ **then**
      $\mathbf{t}_j(5:7) = END, \mathbf{t}_j(8:10) = START$
    end
    else
      $\mathbf{t}_j(5:7) = START, \mathbf{t}_j(8:10) = END$
    end
  end
end

proposed approach is linear to the number of points within the scene.

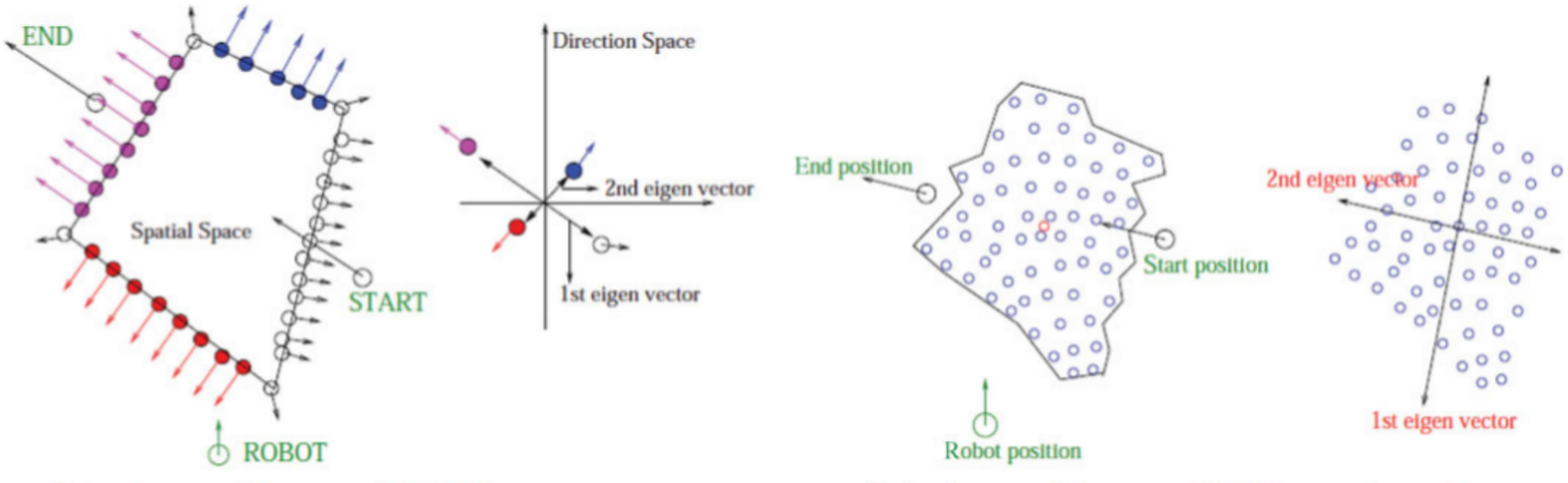
3.2. Gap traversability

Following the stage of gap detection, we analyse the shape of gap contours in order to address the following issues:

- Gap traversability: Determining whether the robot can cross over the gap considering its dimensions.
- Gap traversal path: If the gap is traversable, what are the START and END poses that the robot should reach for traversing the gap.

Our approach to address the above issues resides in using two alternative feature spaces. Afterwards, on either of these two feature spaces, we apply Principal Component Analysis (PCA) and elaborate on the traversability of individual gaps. The two feature spaces are constructed either from the contour orientation using the Normal Principal Component Analysis (NPCA) method[17,18] or from the spatial distribution of the interior of the detected gap point cloud. In the sequel, we will discuss both of these feature space extraction methodologies and compare their performance in Section 4.7. The overall traversability analysis is given in Algorithm 2.

3.2.1. Contour-based feature extraction. Given an individual gap point cloud G_j, we first extract the convex polygon Con_j of this planar point cloud. Afterwards, we uniformly



(a) According to NPCA on contour. (b) According to PCA on interior.

Fig. 5. (Colour online) Gap traversability analysis.

sample the convex contour Con_j and obtain Ct_j, from which we extract the directional feature space by the estimation of the normal vector at each point along the contour Ct_j. The main idea is explained in detail in our previous work,[18] where NPCA was formulated and proposed for the rotation normalization of 3D objects. Here instead NPCA is applied in the 2D domain for the purpose of feature extraction as exemplified in Fig. 5(a).

3.2.2. Interior-based feature extraction. Alternatively, we may perform feature extraction in the original space, taking into account the non-uniform distribution of acquired 3D points due to the laser mode of sensing. To alleviate this limitation of non-uniformity, we down-sample the point cloud according to a neighbourhood size. First we build up a Kd-tree[19] for each gap point cloud. We terminate the branching of the tree when we reach at a leaf size below a distance threshold ϵ. The leaf size is defined as the maximum distance of a point from the centroid of its cluster. With this algorithm, we would like to make the point cloud G_j uniform. It is true that if the *maximum of minimum distance between any two points* within G_j is τ, then the leaf size of the tree should be $\epsilon = \tau$ to make G_j uniform. Since G_j is generated from the image domain according to contour C_j (see Algorithm 1), this value of τ can be evaluated from the maximum radial value within an individual gap point cloud as described by the Eqs. (2)–(5). Let us denote the maximum value of the radial index l_i (see Eq. (3)) for the jth gap within the binary image as $\max(l_i) = L$. Then the value of τ should be derived according to the following formula:

$$\tau = \left(\delta_r^{L-1} - \delta_r^{L-2}\right) r_d. \tag{6}$$

3.2.3. Traversability analysis. After extracting the feature space (either contour or interior), we employ PCA to derive the shape characteristics of the detected gap. In order to assess traversability, we assign $\mathbf{e}_{\text{opt}}$, the optimal eigen-vector as the first principal direction for the contour-based feature space and the second principal eigen-vector for the interior-based feature space. This can be better apprehended from Fig. 5 and Algorithm 2. Afterwards, we project all the original gap points G_j onto the optimal principal direction $\mathbf{e}_{\text{opt}}$ for the evaluation of START and END poses. Moreover, let us say that the set of projected points are denoted as $G_j^{'} = \{g_{j,k}^{'} = \mathbf{g}_{j,k}\mathbf{e}_{\text{opt}}^T\}$, where $k = 0, 1, \ldots, N_j$ for the jth gap and N_j is the number of points in the uniformly sampled point cloud G_j. We define $\max_{k,l} ||g_{j,k}^{'} - g_{j,l}^{'}||$ as the maximum distance between any two points $g_{j,k}^{'}$ and $g_{j,l}^{'}$ from the set $G_j^{'}$. The value that is finally assigned to len_j reflects the length of the gap if it were traversed from the narrowest side. Essentially, this value is used to condition the traversability of the corresponding gap, considering the mobility capabilities of the robotic vehicle. In particular, the gap G_j is deemed as untraversable in the following cases:

1. If len_j is more than a threshold d_r, which is based upon the length of the robot footprint.
2. If estimated START or END poses reside very close to another gap G_k or close to a positive obstacle.

Table II. Parameters.

Robot and environment-specific parameters	Laser-specific parameter	Algorithmic-specific parameters
Plane threshold: Pl_{th}	Increment of intervals between two consecutive radius values: δ_r	Number of morphological operations: n_o
$x_{\min}$		resolution of r: I_{height}
$z_{\max}$		resolution of θ: I_{width}
R		

3.3. Parameters

In this section we highlight those parameters of the proposed approach that are more relevant to the overall performance. We may roughly classify the set of parameters into three groups, as described in the following, and summarized in Table II.

1. **Robot and environment-specific parameters:**
   - Plane threshold (Pl_{th}): The value of this parameter depends upon two factors, namely, the expected terrain roughness, and the degree of roughness that the vehicle can tolerate without adapting its articulating components.
   - $x_{\min}$: The minimum range along the x-direction from where we are bounding the sensing box.

Fig. 6. Snapshots from the tunnel car accident scenario at Montelibretti's firefighters school in Italy.

- $z_{\max}$: The maximum range along the z-direction that could be due to a positive obstacle. This parameter is fixed by the vehicle's height. The proposed methodology does not concentrate on positive obstacle detection. Any 3D point above the robot supporting plane is considered as a positive obstacle. If we assume that a 3D point **p** is above this threshold $z_{\max}$, and there is no other 3D point **q** at the nearby (x, y) coordinate which is having less z value, then it can be easily understood that the point **p** will not affect the traversability of the robot.
- R: Maximum sensing radius. The sensing box boundaries along the x and y-axes are also fixed by this parameter.

2. **Laser-specific parameters:**
   - δ_r: Increment in intervals of consecutive r values, as explained in Fig. 3.
3. **Algorithmic-specific parameters:**
   - n_o: Number of erosion (and dilation) operations. This parameter depends on laser specifications and the corresponding δ_r value.
   - Resolutions of r and θ are I_{height} and I_{width} respectively. These parameters define the number of different r and θ values within the image plane. In Section 3.1, we have used the parameter θ_{int} that is equal to π / I_{width}. Again, given the values of δ_r and I_{height}, the individual r values can be calculated using Eqs. (3)–(5).

4. Experiments

In this section we evaluate the proposed gap detection and traversability analysis approach within USAR and outdoor, urban scenarios. Our aim is to evaluate the performance of the

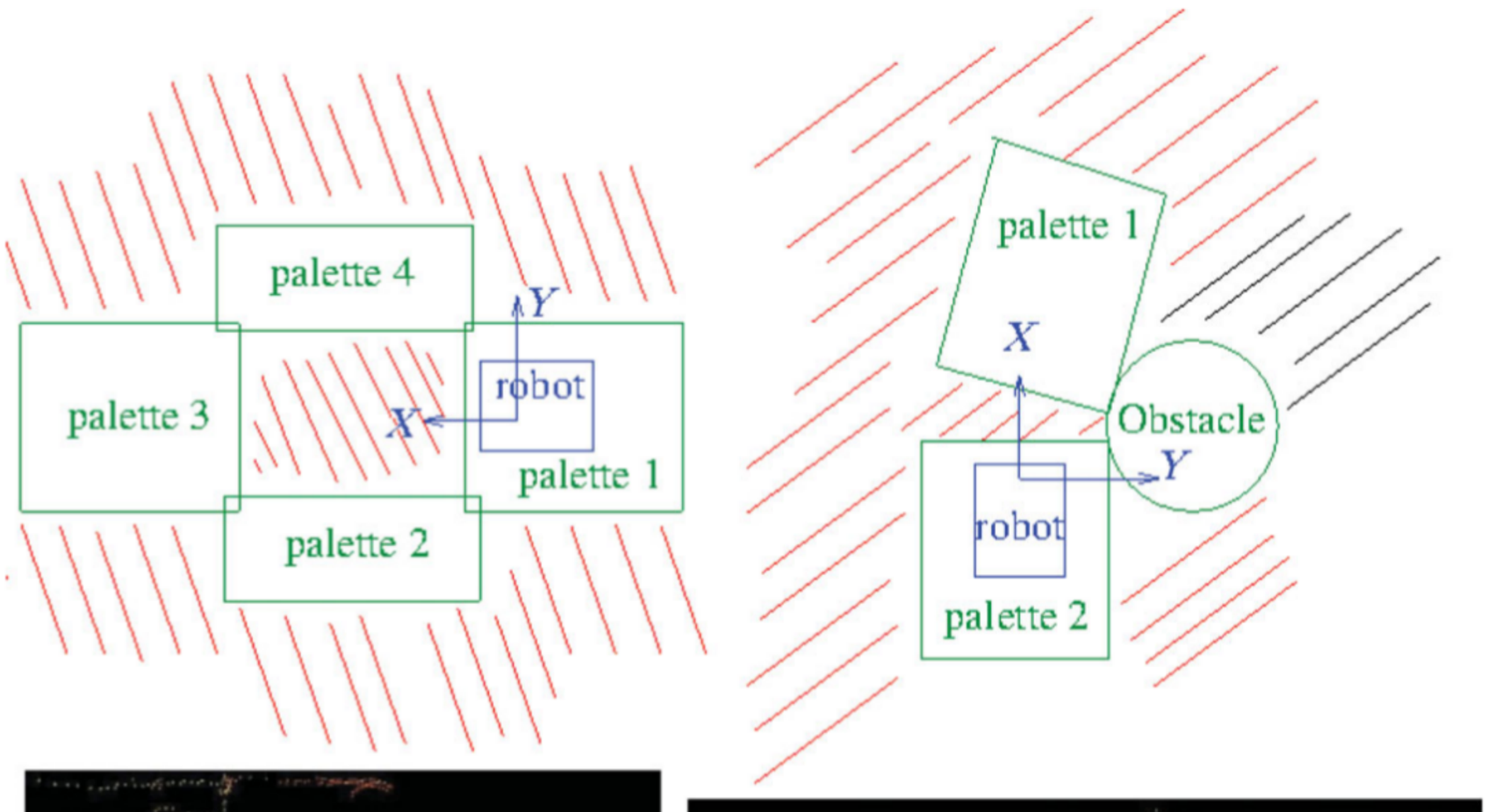
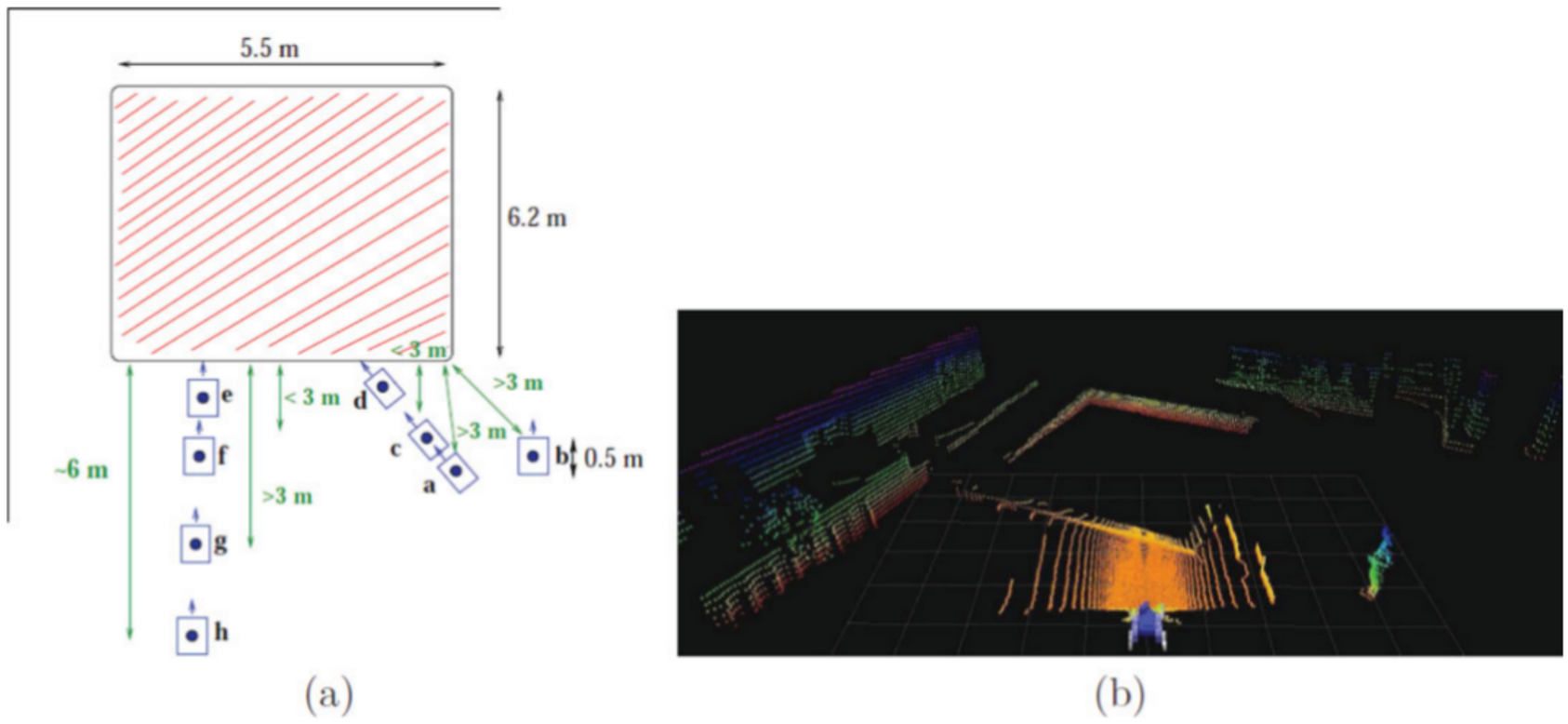


Fig. 7. (Colour online) Top row: Schematic top-down views of two experiment scenarios. Red lines correspond to negative obstacles, and black (straight) lines signify occlusion due to a positive obstacle. Bottom row: Top-down views of acquired 3D point clouds within the corresponding scenarios.



(a) (b)

Fig. 8. (Colour online) Scenario 3: (a) Schematic representation of the scene. Blue rectangle and arrow denote the robot position and orientation, gaps are signified through red lines, while the distance from the robot to the gap is shown in green. (b) Point cloud acquired from position c shown in coloured coding. Yellow signifies the minimum z-value, while blue signifies the maximum z-value.

proposed methodology under varying parameter settings that assist in deriving its limitations and encountering its failure cases which finally provides us with the optimal modes of operation. The experiments that we have conducted further demonstrate the performance of our method with respect to different viewing angles and locations of the vehicle within the scene.

4.1. Evaluation scenarios

4.1.1. Tunnel scenario. The first two experiments were performed within a realistic tunnel car accident site (depicted in Figs. 1 and 6). These experiments were conducted for the purpose of end user's evaluation of robotic platform used within the EU FP7 IP **NIFTi** project (www.nifti.eu) that concerns Natural Human Robot Cooperation in Dynamic Environments.

We have evaluated our approach within two variations of a scene as depicted in Fig. 7. These two scenarios correspond to two different situations where the negative obstacle is too close to the robot. The 3D point clouds corresponding to each scenario are shown at the bottom of Fig. 7.

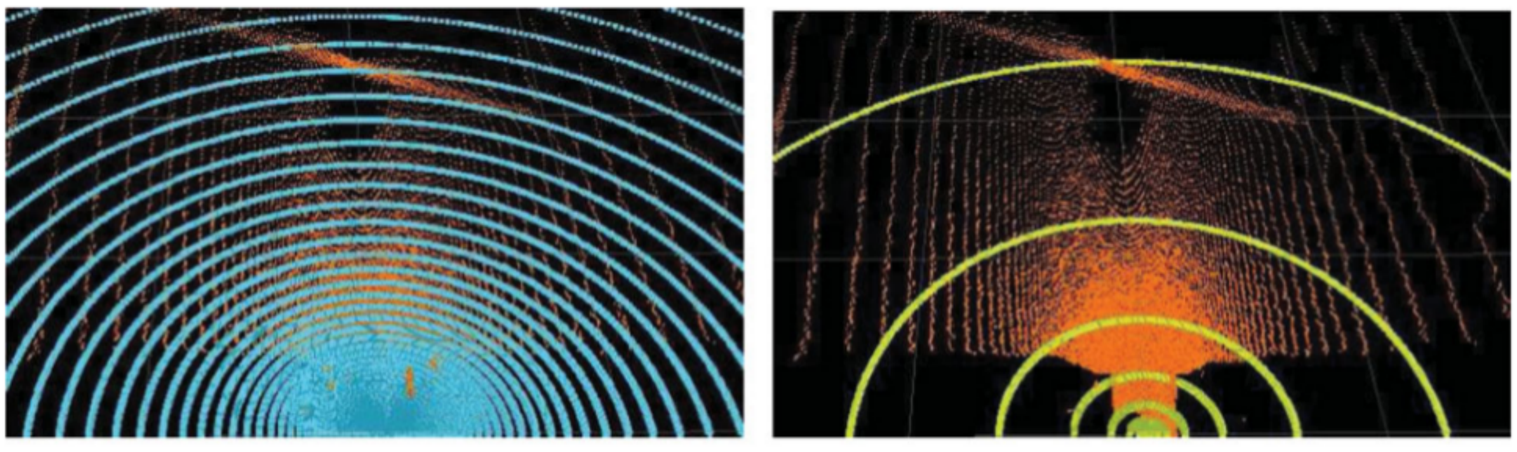
1. The first scenario corresponds to a constrained situation wherein the robot is unable to move in any direction.
2. The second scenario allows a single moving direction (corresponding to a small and traversable gap in the foreground). The added complexity of this scenario in comparison to the first scenario is the presence of a positive obstacle (barrel) that occludes a portion of the scene.

4.1.2. Outdoor scenario. In Fig. 8(a) we provide the schematic diagram of the outdoor scenario (scenario 3) and the robot positions ($a, b, \ldots, g, h$) with robot-pose direction (depicted as a blue arrow). In Fig. 8(b), we show the corresponding point cloud acquired from the robot at position c to give a view of the surrounding area. This scenario is used to evaluate the performance of the proposed methodology under different viewing angles and robot positions.

4.2. Effect of parameters

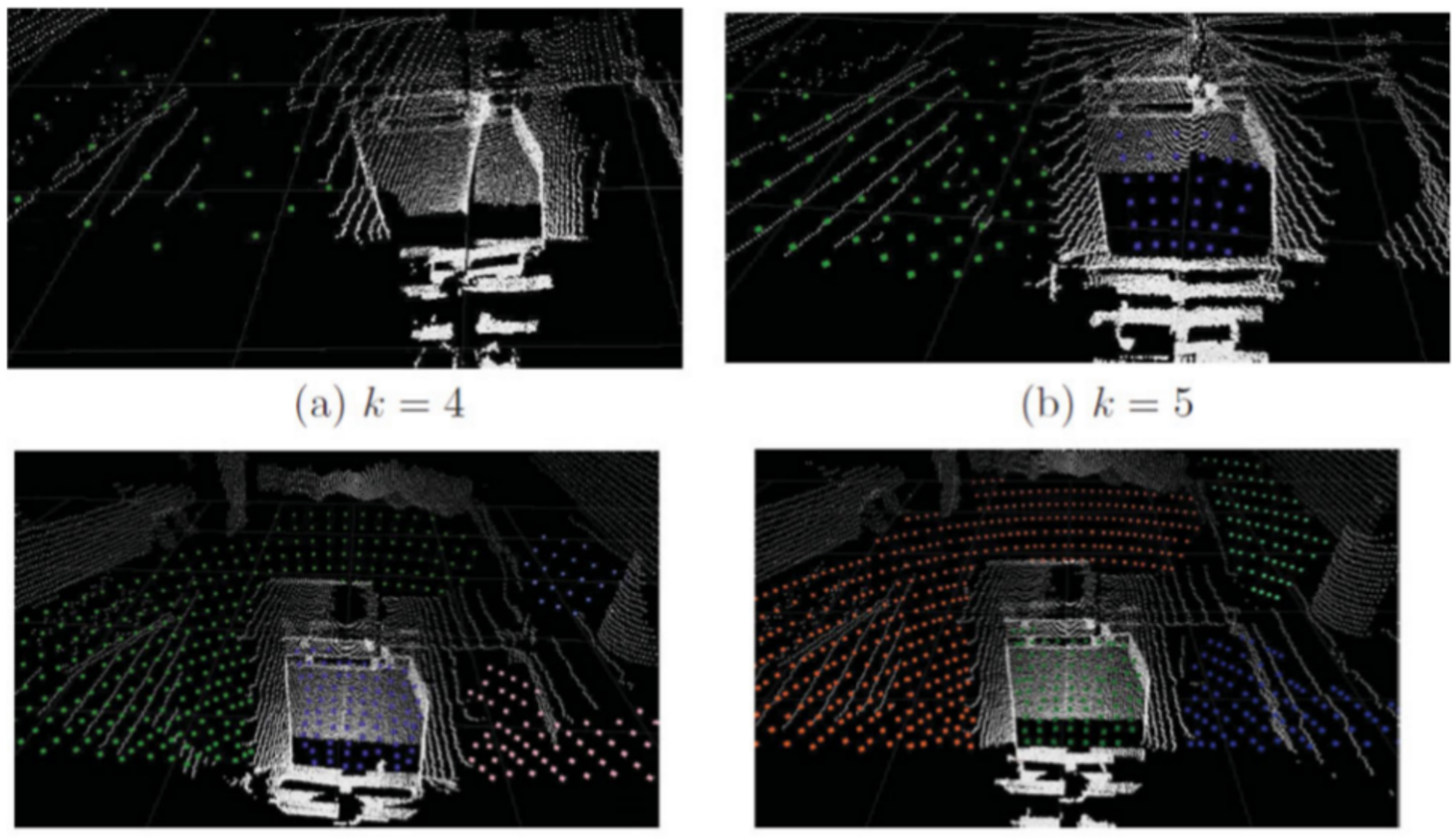
The robot and environment-specific parameters may vary according to robot specifications. For the specific robot used in these experiments, we have fixed the values of the corresponding parameters (in metres) as $Pl_{\text{th}} = 0.1$, $x_{\min} = 0.6$, $z_{\max} = 2.0$, $R = 3.0$.

4.2.1. Parameter δ_r. The value of δ_r should generally be greater than one and according to Larson *et al.*,[12] the optimal value is $\delta_r = 2$. In Fig. 9, we show two results with respect to two different δ_r values, namely, $\delta_r = 1.1$ and $\delta_r = 2$. The point cloud (shown in blue in Fig. 9(a) and yellow in Fig. 9(b)) corresponds to the grid of the image plane.



(a) $\delta_r = 1.1$ (b) $\delta_r = 2.0$

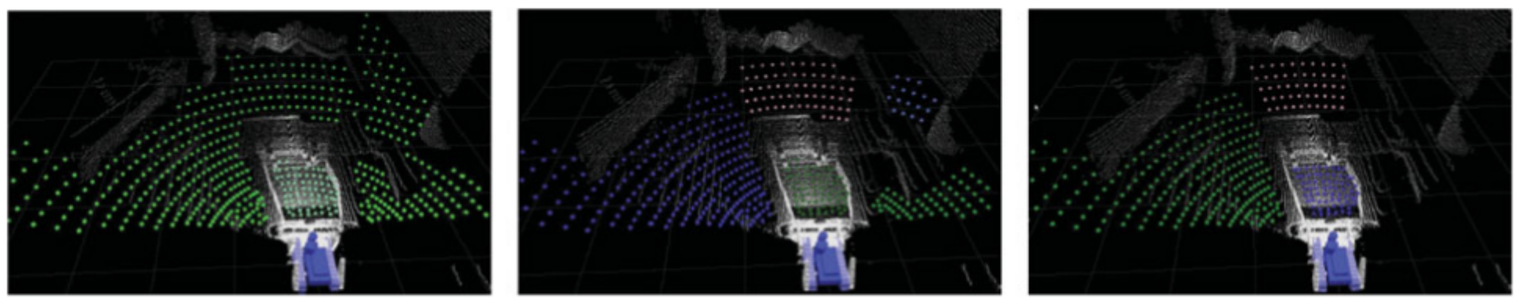
Fig. 9. (Colour online) Effect of the laser parameter δ_r: The blue and yellow pixels in (a) and (b), respectively, show the generated point cloud from the image plane, while orange pixels show the original point cloud. It can be seen that for the laser used in these experiments, $\delta_r = 2.0$ is not the preferred setting in contrast to the approach followed within.[12]



(a) $k = 4$ (b) $k = 5$

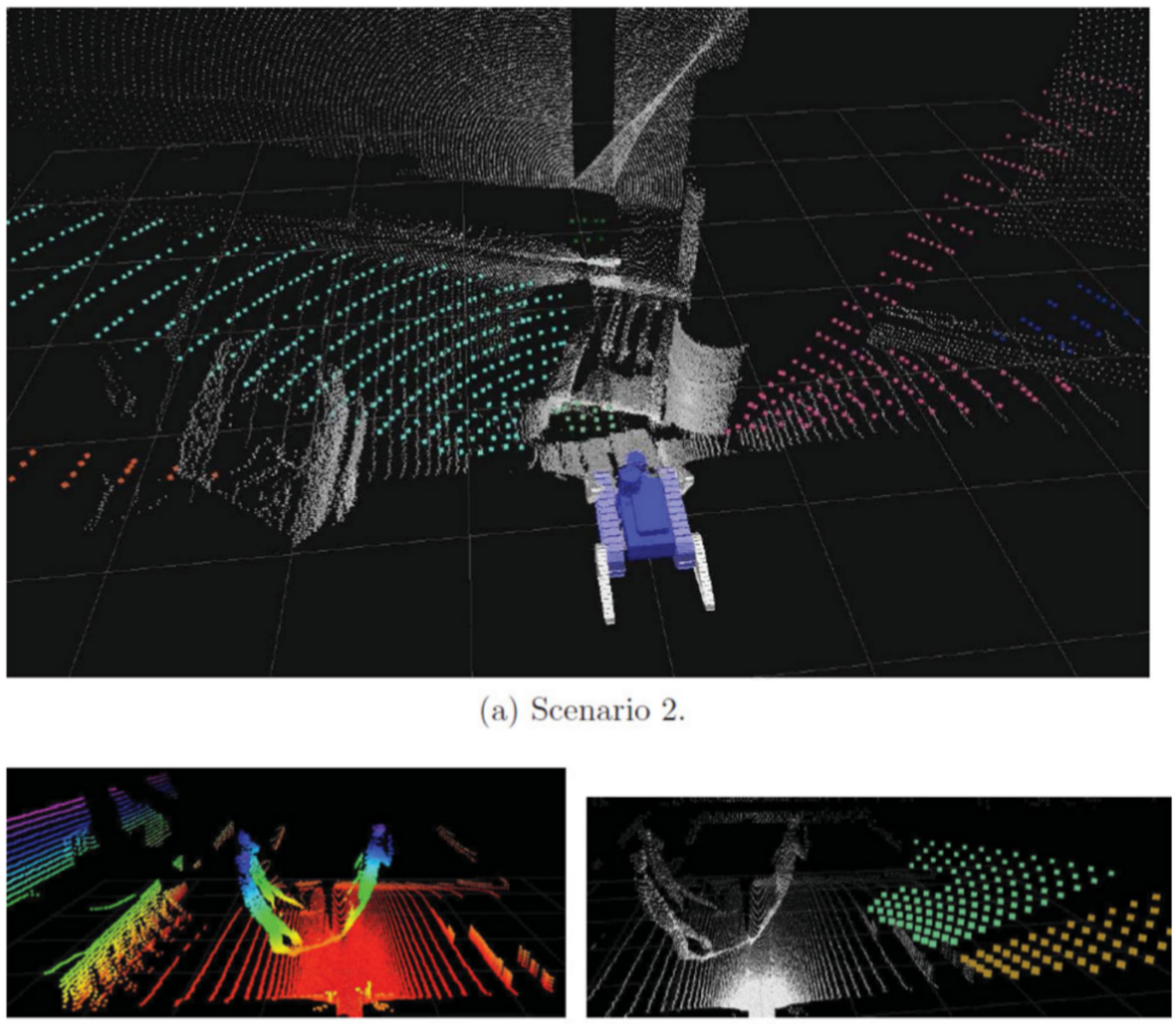
(c) $k = 6$ (d) $k = 7$

Fig. 10. (Colour online) Effect of resolution parameter k by keeping parameter n_o fixed. Resolution is defined as $I_{\text{width}} = I_{\text{height}} = 2^k + 1$.



(a) $n_o = 0$. No morphological operation. (b) $n_o = 1$. (c) $n_o = 2$.

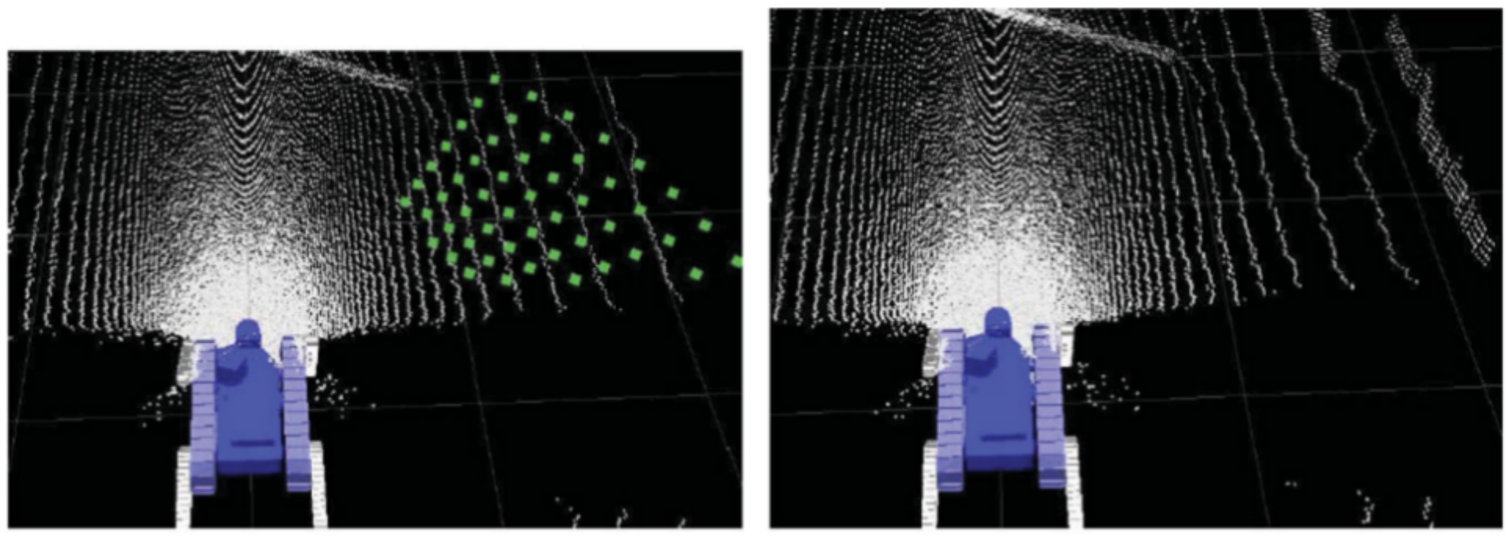
Fig. 11. (Colour online) Effect of morphological parameter n_o by keeping the resolution parameter fixed at $k = 6$; with no morphological operation (a) all the gaps are detected but not grouped; with too many morphological operations (c) some gaps are not detected.



(a) Scenario 2.

(b) Scenario 3. (c) Scenario 3. Gap-detection result.

Fig. 12. (Colour online) Positive obstacle occlusion.



(a) Case 1. Gap detected but in reality there is no gap.

(b) Case 1. At the same location no gap is detected.

(c) Case 2. Gap detected but in reality there is no gap.

(d) Case 2. At the same location no gap is detected.

Fig. 13. (Colour online) Failure cases: occasionally gaps may be detected, although they correspond to false positives.

Here we show the result when the optimal δ_r value for our particular laser sensor is used, namely, $\delta_r = 1.1$, where the interval between two consecutive radial point acquisitions most reliably resembles the radial interval with respect to the original point cloud (shown in orange). Therefore, in the remaining set of experiments we have set $\delta_r = 1.1$. In general, δ_r can be fixed according to the specifications of the LIDAR sensor that is used.

4.2.2. Resolution parameters. Finally, we are interested to see the results with respect to the three parameters of the algorithm, namely, n_o, I_{height} and I_{width}. Without loss of generality, we set $I_{\text{height}} = I_{\text{width}} = 2^k + 1$. Therefore, a small variation in k will result in large variation in the resolution of both r and θ that has a direct effect on the performance of the proposed methodology, as will be shown.

In examining the effects of the parameters n_o and k on the gap detection result, we choose the first scenario as shown on the left of Fig. 7. Initially, we fix the parameter $n_o = 1$ to examine the dependence of the result with respect to parameter k. It can be derived from Fig. 10 that after a certain threshold ($k = 6$) the performance increase is trivial, if present. We concluded that the benefit in terms of accuracy in gap perception for values of k higher than $k = 6$ was not significant enough to compensate for increase in computational cost (see Section 3.1.4) that has been kept very low in order to allow for real-time performance (≤ 50 ms, with $k = 6$). The overall time performance is discussed in Section 4.8.

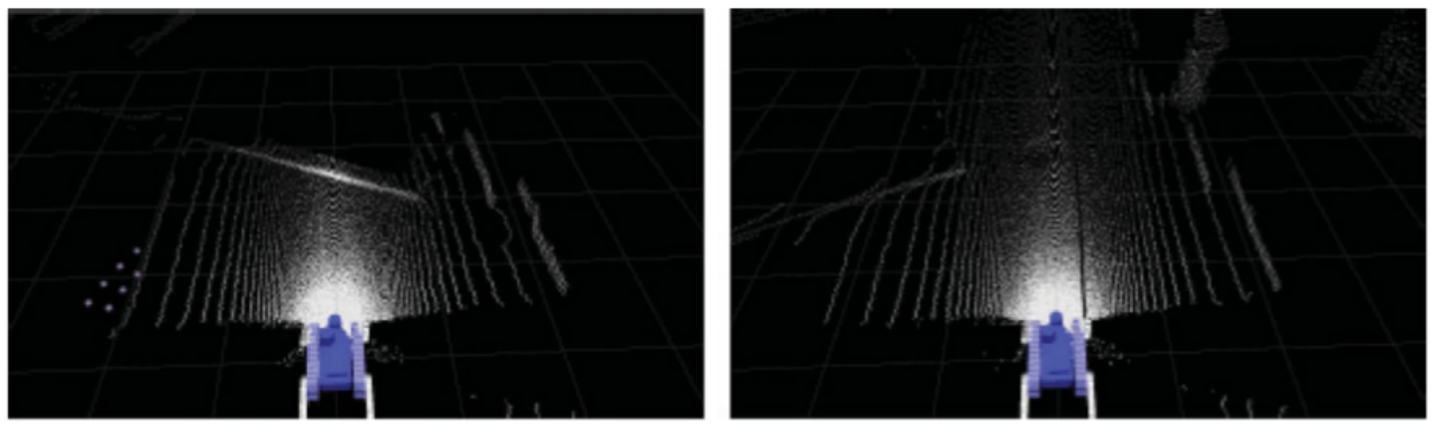
4.2.3. Number of morphological operations n_o. In the next stage of our evaluation, we fix the resolution parameter to the optimal value ($k = 6$) and vary the n_o parameter within the interval $[0, 2]$ (since this range is sufficient to explain the effect). Figure 11 shows the effect of morphological operation with respect to scenario 1.

As can be observed from Fig. 11(a), with no morphological operation only two gaps are detected (with light green and dark green colours), one just in front of the robot and the other in its surroundings. On the other hand, when the number of morphological operation is increased to two as shown in Fig. 11(c), some gaps are not detected. Finally, when the number of morphological operations is one, as can be seen in Fig. 11(b), all the gaps are consistently detected. From these results, one can easily conclude that morphological operation helps to segment different gap regions, although the perceptual information maybe be distorted above a certain number of operations.

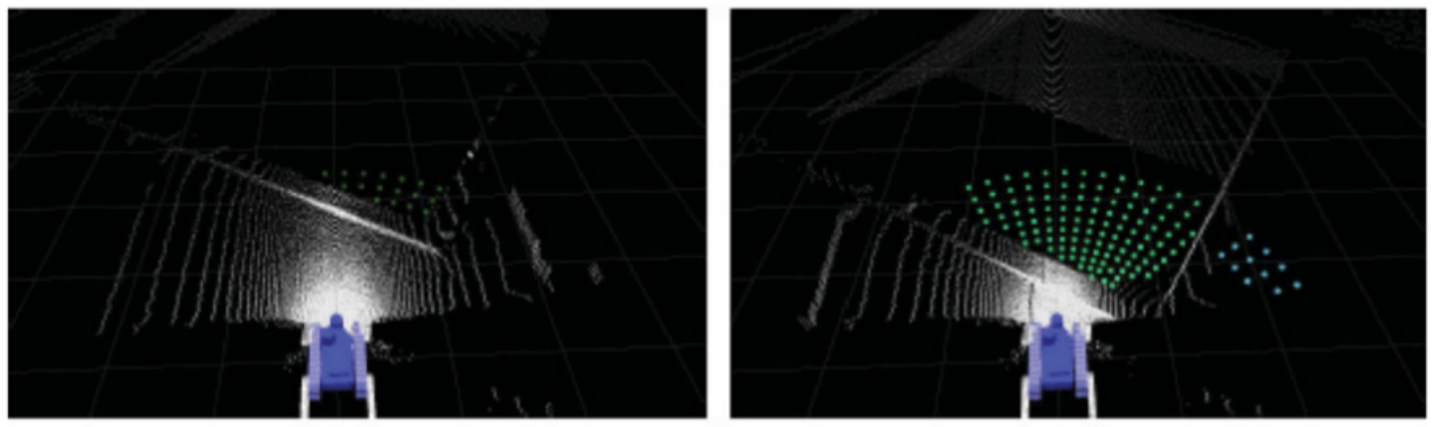
The grouping of gaps is advantageous, since it helps to study the traversability with respect to each gap and reason based upon these results, as described in Section 3.2.3. Therefore, it should be estimated according to the robot mobility capabilities. In our experiments, we finally consider a single morphological operation, as this option provided the most stable results.

4.3. Positive obstacle occlusion

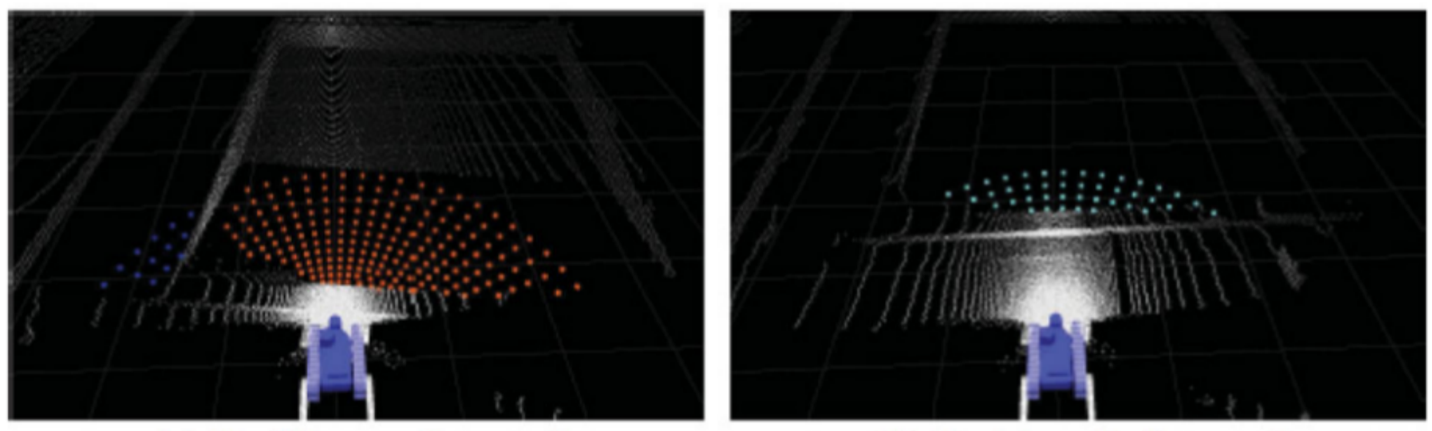
As shown in Fig. 7 for the second scenario, in the presence of a positive obstacle in front of the robot, the absence of points behind this positive obstacle cannot be used as a clue for the presence of negative obstacles or gaps. We test this scenario using the optimal parameter setting as described in Section 4.2. Figure 12(a) shows the result of not detecting the



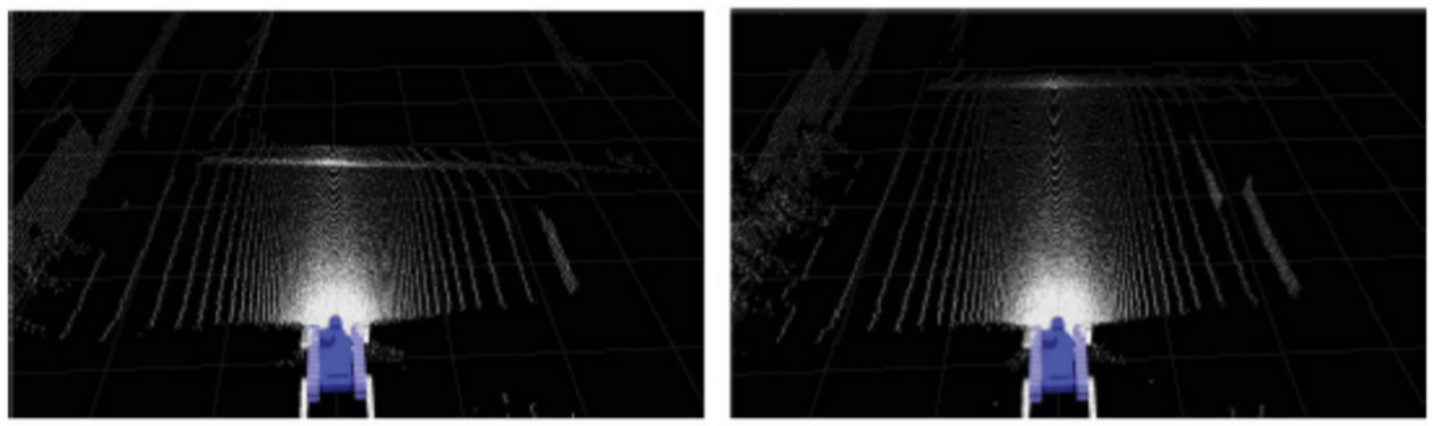
(a) Position a: not detected. (b) Position b: not detected.



(c) Position c: detected. (d) Position d: detected.



(e) Position e: detected. (f) Position f: detected.



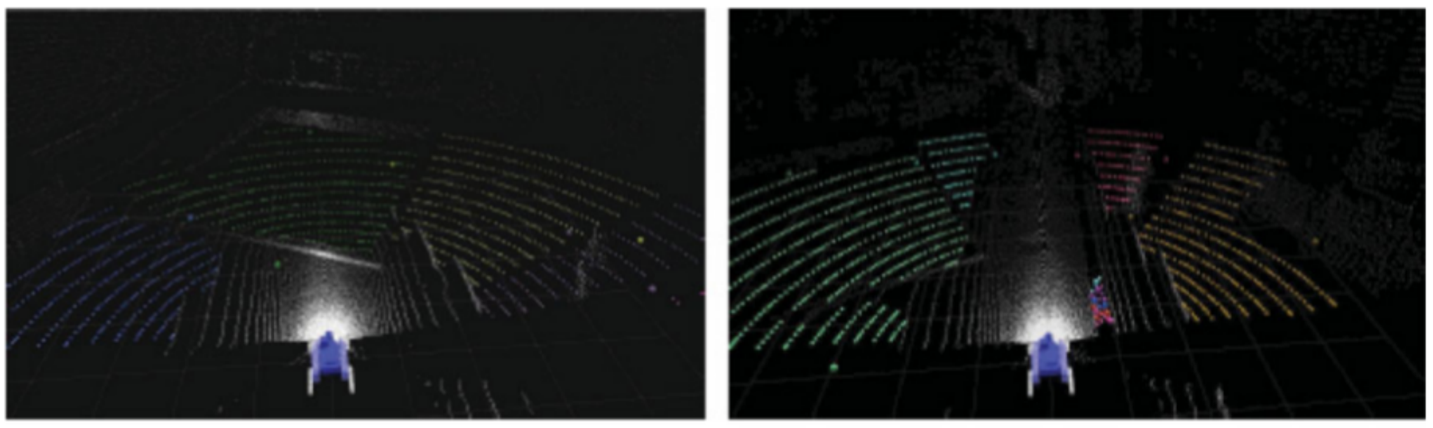
(g) Position g: not detected. (h) Position h: not detected.

Fig. 14. (Colour online) Gap detection results within the outdoor scenario: Different gaps are coded in different colours and white point cloud corresponds to the original point cloud. Fixed parameter set: $\delta_r = 1.1$, $n_o = 1$, $Pl_{\text{th}} = 0.1$. Variable parameter set: $I_{\text{height}} = 2^6 + 1$, $I_{\text{width}} = 2^5 + 1$, $R = 3$ m. Occasionally the gaps are not detected due to the small radius R of the sensing area.

absence of points (occluded by positive obstacle) as a gap. Moreover, within the third scenario, there is one instance where two human subjects move in front of the robot as shown in the Fig. 12(b). The absence of points beyond these positive (moving) obstacles should not be labelled as gaps. As shown in Fig. 12(c), gaps are detected only when the absence of points is not caused by positive obstacle occlusion.

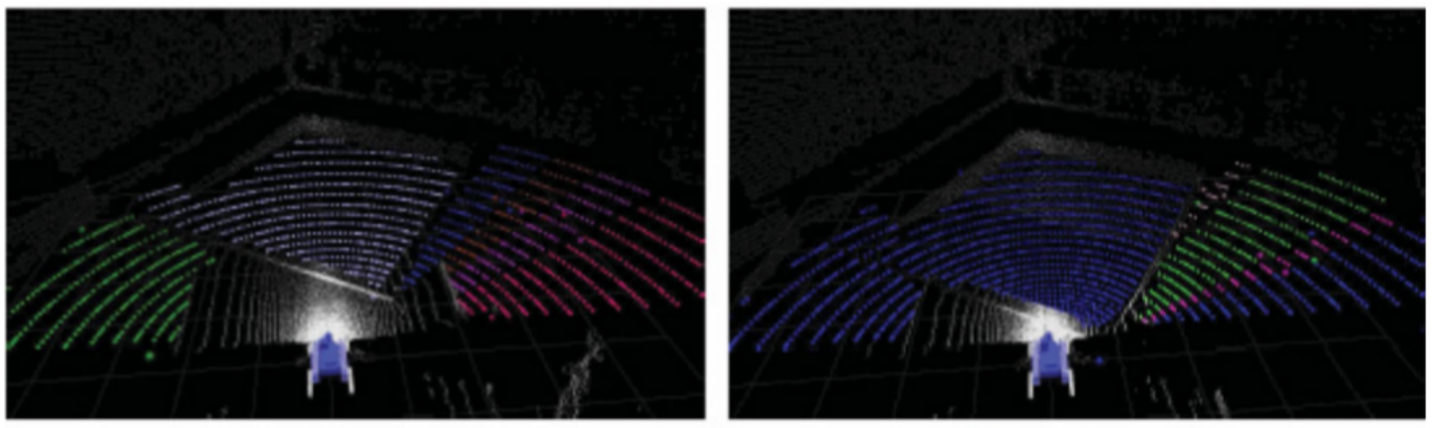
4.4. Comparison with earlier approaches

As explained in Section 2 and Table I, there are four earlier works[4,7,11,12] that correspond to the same sensor allotment, that is, gap detection using the LIDAR data. The methodology that we propose differs from these works in two perspectives, namely, the gap detection approach and the consecutive traversability analysis that can support high-level reasoning during the path planning process. Our gap detection methodology incorporates the following two key features: First, the non-uniform sampling of the robot supporting plane as expressed by Eqs. (2)–(5) assists in tuning the laser-dependent parameter as described in Section 4.2.1. Second, we employ image contour analysis in combination to mathematical morphology that allows the grouping of individual gaps for traversability analysis. And last but not least, we perform traversability analysis of individual gaps that allows for high-level traversability analysis, a concept that herein is explored for the first time. In this direction, we apply PCA in two different feature spaces for the extraction of corresponding traversability information for each individual gap.



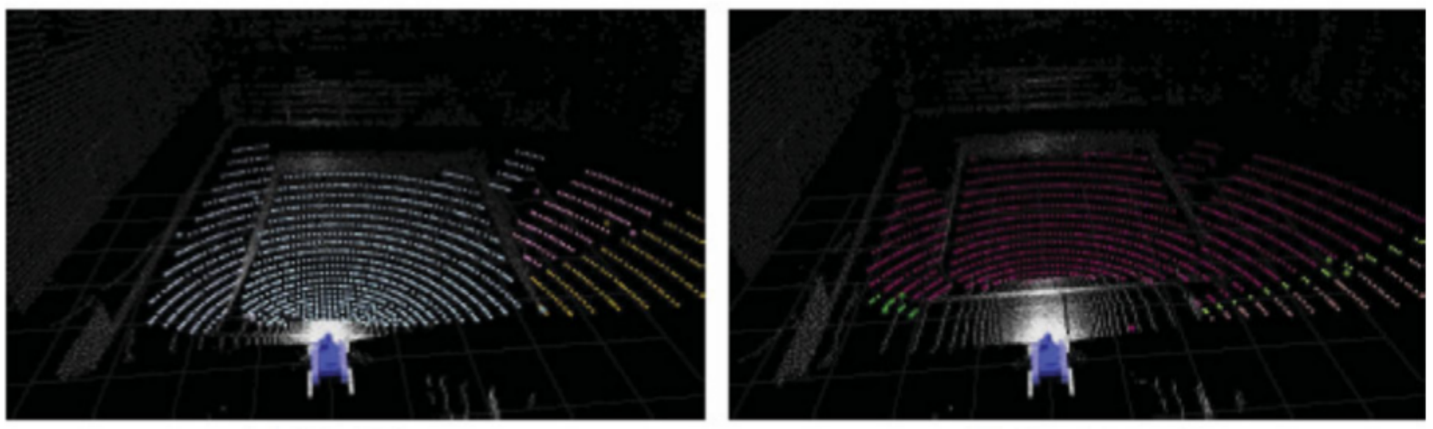
(a) Position a.

(b) Position b.



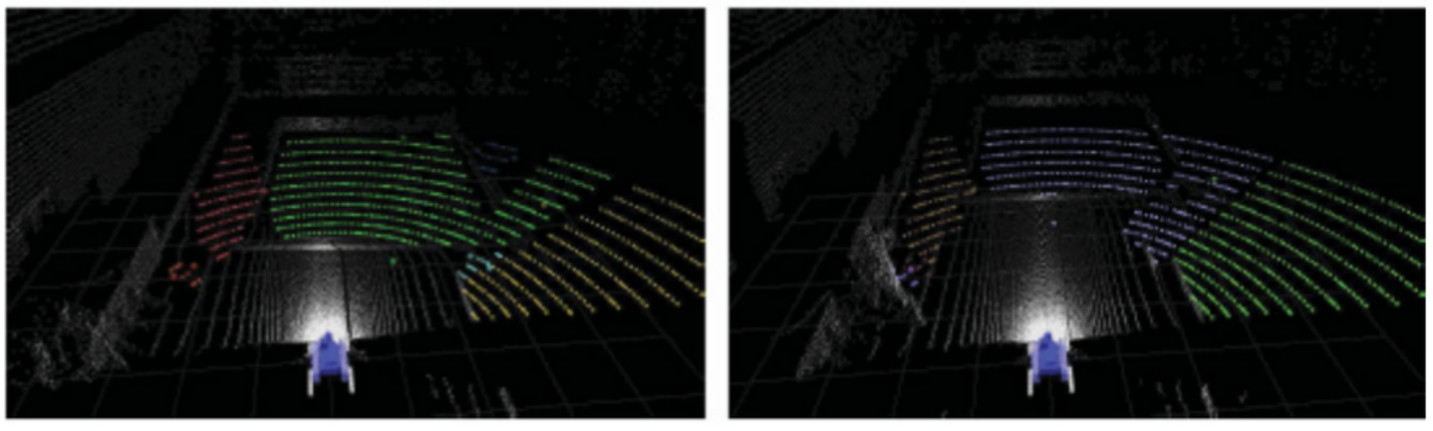
(c) Position c.

(d) Position d.



(e) Position e.

(f) Position f.



(g) Position g.

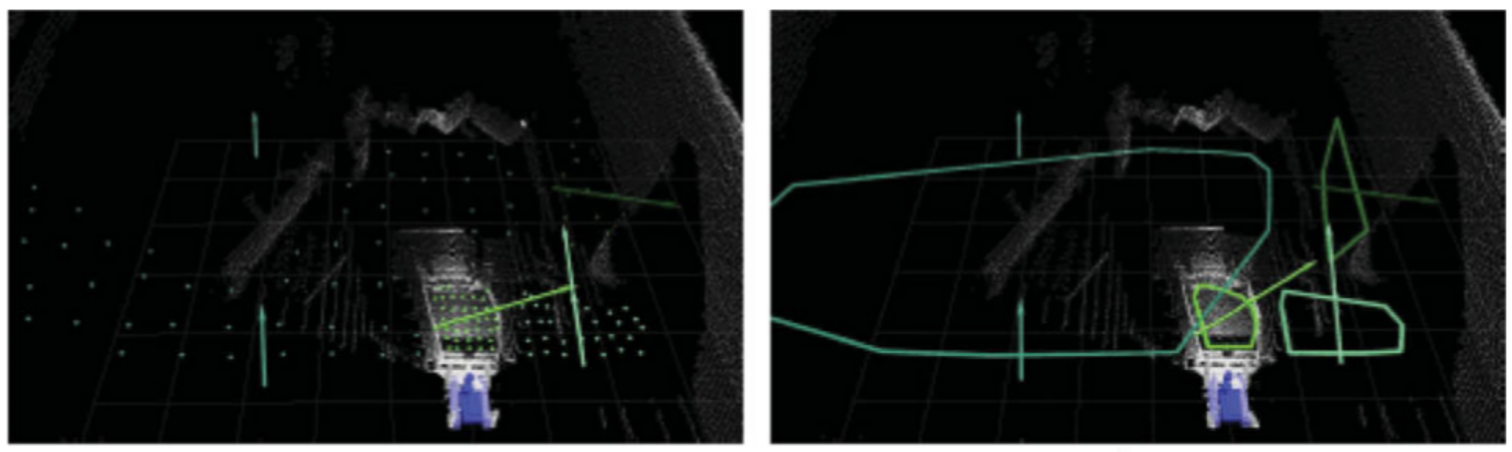
(h) Position h.

Fig. 15. (Colour online) Gap detection results with a radius of 10 m within the outdoor scenario. Different gaps are coded in different colours and the white point cloud corresponds to the original point cloud. Fixed parameter set: $\delta_r = 1.1$, $n_o = 1$, $Pl_{\text{th}} = 0.1$. Variable parameter set: $I_{\text{height}} = 2^6 + 1$, $I_{\text{width}} = 2^6 + 1$, $R = 10$ m. Gaps are detected in all cases, since the sensory radius R is sufficiently large.

(a)

(b)

Fig. 16. (Colour online) Snapshots of the outdoor scenario scene.



(a) Interior (b) Contour

(c) Interior (d) Contour

(e) Interior (f) Contour

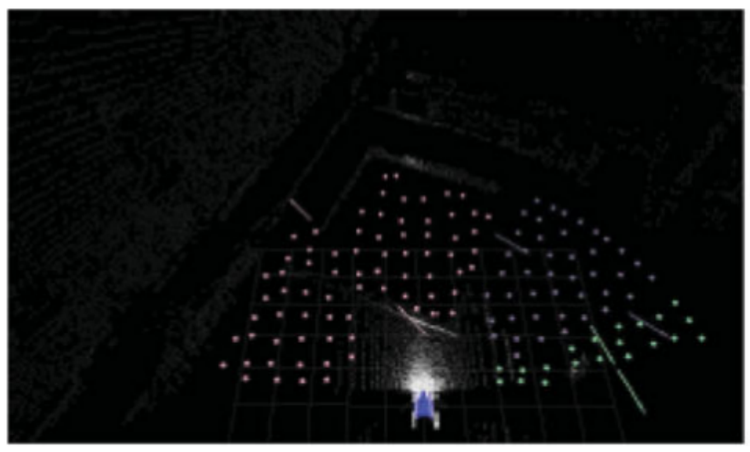
(g) Interior (h) Contour

Fig. 17. (Colour online) Scenario 1: Gap traversability analysis results using conventional PCA (interior) and NPCA (contour).

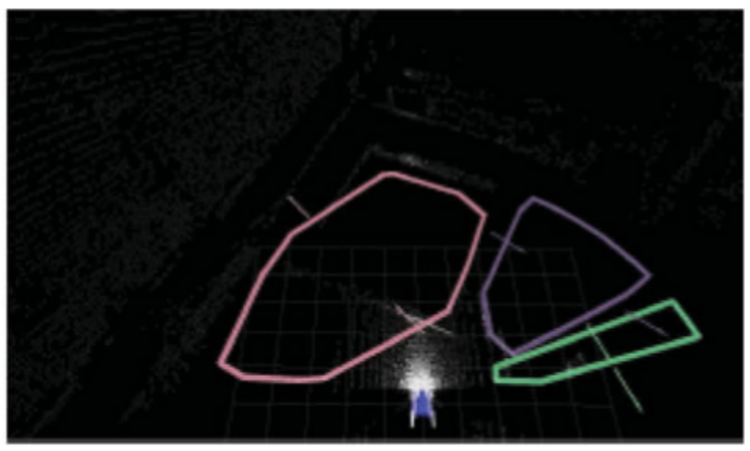
Unfortunately, earlier approaches for gap detection cannot be easily reproduced due to limited descriptions in refs. [4, 11, 12] in terms of learning or adjusting the internal and external parameters. In contrast, we extensively discuss the parameters of the proposed algorithm in order to render the proposed methodology eligible for testing on different platforms by suitably adjusting the corresponding parameters as described in Section 4.2.

4.5. Failure cases

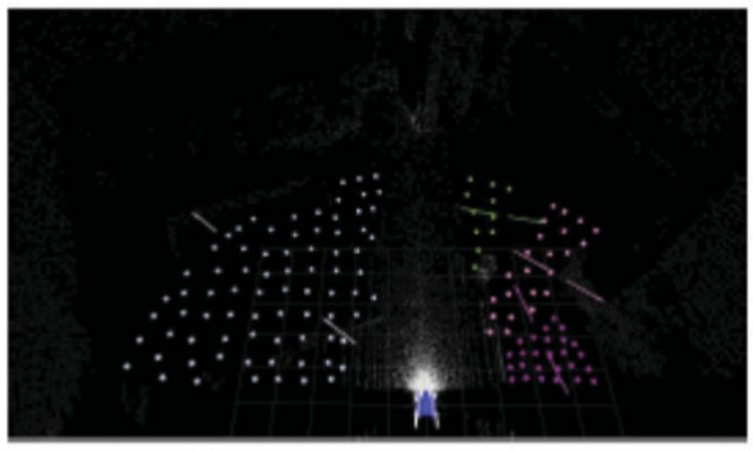
In Fig. 13, we give an example of false positives detection using our gap detection methodology. Two cases are shown (respectively in two rows). For each case, at the same location two point clouds are generated. Each row shows the result of the proposed algorithm. For these results, we obtain that the proposed methodology occasionally assess the presence of a gap although there is no real gap. We have observed that this behaviour is often attributed to the susceptibility of the 3D pose estimation of the robot to errors that results in inconsistency for the underlying supporting plane computation. This is an expected result according to the formulation of the proposed gap detection approach, which relies on a robust 3D pose estimation process. However, the confidence of the 3D pose estimation can be easily quantified, which in turn allows for weighing the certainty in the gap detection result. As far as safe robot navigation is concerned, a high false positive ratio may result in a relatively *intimidating* robot behaviour; however, it is guaranteed that *if a gap really exists*, then the algorithm will detect the corresponding gap so that we obtain a true positive 100%.



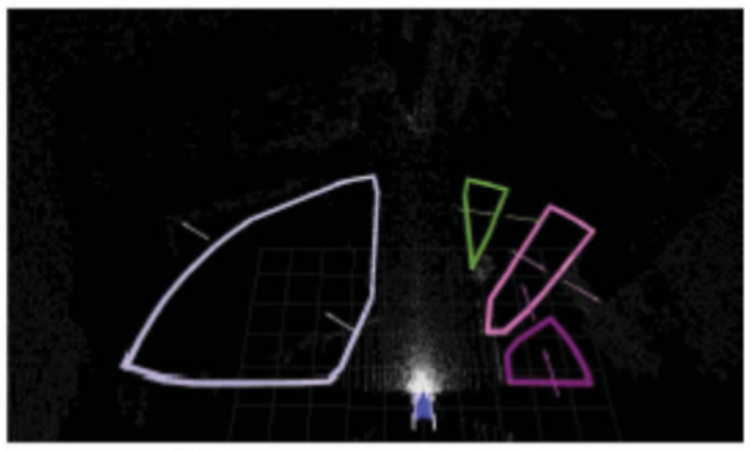
(a) Position a: interior.



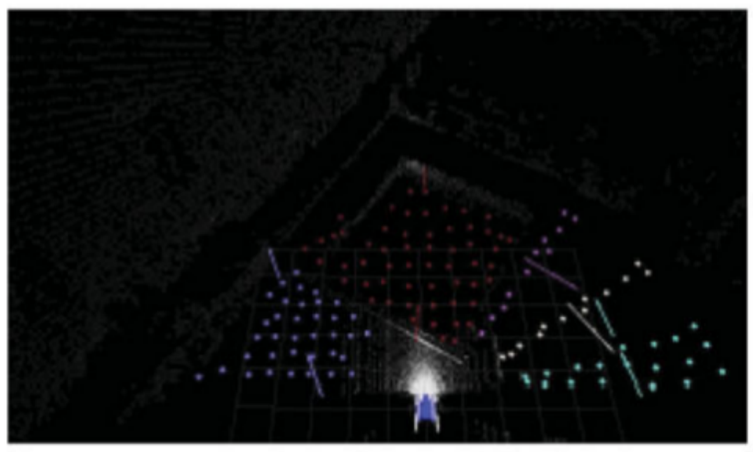
(b) Position a: contour.



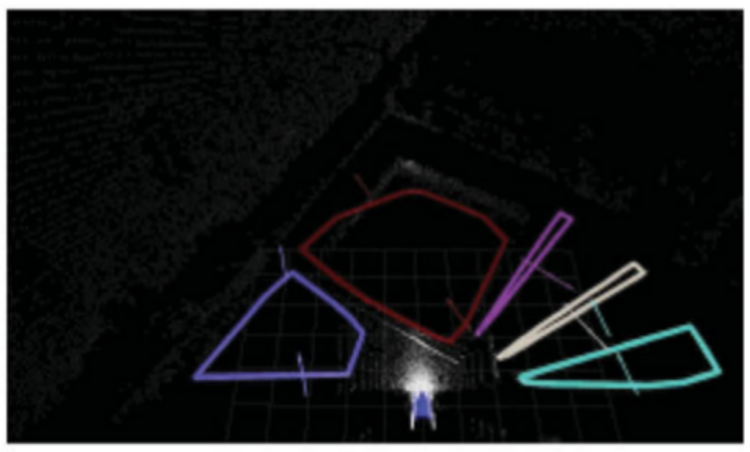
(c) Position b: interior.



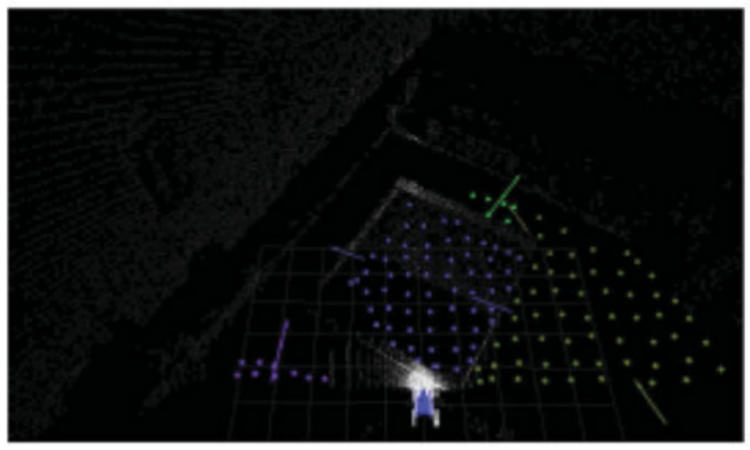
(d) Position b: contour.



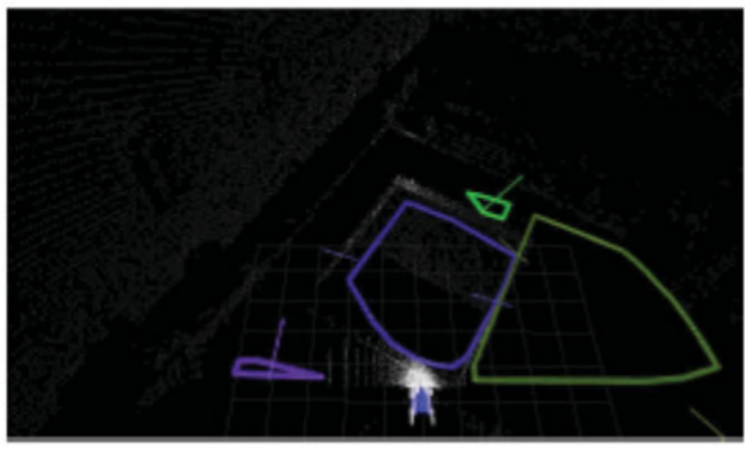
(e) Position c: interior.



(f) Position c: contour.



(g) Position d: interior.



(h) Position d: contour.

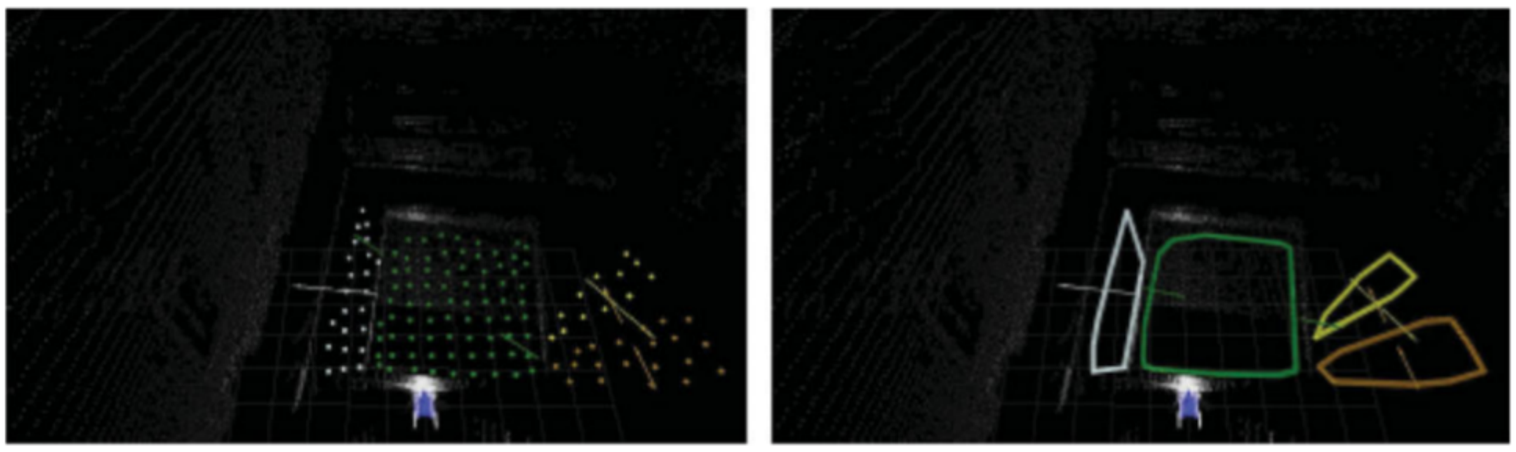
Fig. 18. (Colour online) Scenario 3: Gap traversability analysis results using conventional PCA (interior) and NPCA (contour).

4.6. Outdoor scenario evaluation

In Figs. 14 and 15, we show the results of applying the proposed methodology within the outdoor scenario as described in Section 4.1.2 and shown in Fig. 16. It can be seen that the proposed approach is robust to changes in the viewing angle and distances. Occasionally, as has been already discussed at Section 4.5, false positives may appear. This is almost always attributed to the sparsity of acquired 3D points due to increased distance from the robotic vehicle. Through a careful look at the acquired point clouds, it can be seen that the regions where there is no laser data (which is not due to obstacle occlusion) should be detected as a gap region. This is an expected result according to what the proposed approach is designed to perform. Therefore, from a complete view of the scene it could be derived that these correspond to false positives but this cannot be assessed solely from a single 3D scene acquisition.

4.7. Results of traversability analysis

In Figs. 17–19, we demonstrate the gap traversability analysis results with respect to different individual gaps as detected within different point clouds. These figures also show the comparison between the two alternative methodologies that we explored for traversability analysis, namely, using either the contour direction information or the interior points of the gap in the original 3D space. As can be seen, both methodologies perform quite similarly and produce good results even if the gaps are not of elliptical or rectangular shapes. In these results we prompt to show the effectiveness of our proposed traversability direction evaluation. We do



(a) Position e: interior. (b) Position e: contour.

(c) Position f: interior. (d) Position f: contour.

(e) Position g: interior. (f) Position g: contour.

(g) Position h: interior. (h) Position h: contour.

Fig. 19. (Colour online) Scenario 3: Gap traversability analysis results using conventional PCA (interior) and NPCA (contour).

not show whether the gaps are traversable or not since when given the START and END poses, it is straightforward to assess the traversability of any gap.

The similar performance that we attest in the two approaches is a way in which we can evaluate the stability of the proposed gap analysis methodology. If the two approaches gave significantly different results, then this would be a strong indication of instability of the approach. As is discussed in previous work,[18] NPCA and Continuous PCA (CPCA) give different results that result in an overall highly complementary behaviour when both direction information and spatial information are used for the purpose of normalizing the rotation of 3D objects. However, in the application that is considered in this paper, namely, 2D gap contour analysis, the inverse behaviour is observed, hence one can choose to perform the analysis of the shape of a contour in any of the two feature spaces and obtain a very similar result.

4.8. Time performance

The experiments reported within this work have been performed by using a system equipped with a 64 bit, Intel-I7 CPU and 7.8 GB memory. The overall time cost (gap detection and traversability analysis) ranges from 7 to 30 ms for an acquired 3D scene, with an average of 15 ms.

5. Conclusions

We have proposed a novel methodology for effective and efficient detection of negative obstacles in the form of gaps,

together with a framework to analyse the gap traversability by considering not only the terrain but also the mobility capabilities of a robotic vehicle. Together with the proposed gap traversability analysis approach that allows for non-binary traversability assessments for the first time, in the present work, we have reported on the various difficulties that are encountered in the perception of negative obstacles and proposed solutions to address each individual problem through extensive experiments within USAR environments and conditions that had not so far been explored.

Through our experiments, we have shown that the problem of gap detection and traversability analysis can be alleviated by suitably employing state-of-the-art signal processing techniques that allow a robotic vehicle to navigate in a safe as well as optimal mode of operation in very adverse and cluttered environments such as those encountered in USAR scenarios. Ultimately, the proposed framework can be used to enhance the performance of robot path planning locally, as it can signify the presence of gaps and suggest the most adequate path plan according to the given robotic vehicle. In particular, the proposed gap perception and traversability analysis could be seamlessly combined with the ability of the robotic vehicle to assess the traversability–mobility of a given 3D terrain automatically as described by Papadakis and Pirri[16] through physics-based optimization. Such an integration of functionalities would allow the UGV to primarily filter out the regions that have been deemed as untraversable gaps and subsequently evaluate the 3D traversability of continuous solid areas.

Acknowledgements

This paper describes research supported by the EU-FP7 ICT 247870 NIFTi project. We would further like to thank the reviewers for their constructive feedback both in terms of improving the quality of the manuscript as well as for stimulating and inspiring our future work.

References

1. P. Bellutta, R. Manduchi, L. Matthies, K. Owens and A. Rankin, "Terrain Perception for Demo III. *Proceedings of the IEEE Intelligent Vehicles Symposium* (2000).
2. Bluebotics, *Mobile robot. No. PCT/EP2011/060937* (BlueBotics SA, Switzerland, Jun. 2011).
3. Intel Corporation, Open source computer vision library, Available at: http://opencv.willowgarage.com/wiki/ (Aug. 2011). (Accessed April 2013)
4. C. D. Crane III, D. G. Armstrong II, R. Touchton, T. Galluzzo, S. Solanki, J. Lee, D. Kent, M. Ahmed, R. Montane, S. Ridgeway, S. Velat, G. Garcia, M. Griffis, S. Gray, J. Washburn and G. Routson, "Team cimar's navigator: An unmanned ground vehicle for the 2005 darpa grand challenge," *J. Field Robot.* **23**(8), 599–623 (2006).
5. C. Dima, N. Vandapel and M. Hebert, "Classifier Fusion for Outdoor Obstacle Detection," *Proceedings of the International Conference on Robotics and Automation* (2004).
6. G. Dubbelmanand, W. van der Mark, J. C. J. van den Heuvel and F. C. A. Groen, "Obstacle Detection During Day and Night Conditions Using Stereo Vision," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (2007).
7. N. Heckman, J.-F. Lalonde, N. Vandapel and M. Hebert, "Potential Negative Obstacle Detection by Occlusion Labeling," **In**: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (2007) pp. 2168–2173.
8. 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," *Int. J. Robot. Res.* **25**(5–6), 449–483 (2006).
9. G.-J. Kruijff, M. Janicek, S. Keshavdas, B. Larochelle, H. Zender, N. Smets, T. Mioch, M. Neerincx, J. van Diggelen, F. Colas, M. Liu, F. Pomerleau, R. Siegwart, V. Hlavac, T. Svoboda, T. Petrickek, M. Reinstein, K. Zimmerman, F. Pirri, M. Gianni, P. Papadakis, A. Sinha, B. Patrick, N. Tomatis, R. Worst, T. Linder, H. Surmann and V. Tretyakov, "Experience in System Design for Human-Robot Teaming in Urban Search & Rescue," *Proceedings of the International Conference on Field and Service Robotics* (2012).
10. J. F. Lalonde, N. Vandapel, D. Huber and M. Hebert," "Natural terrain classification using three-dimensional ladar data for ground robot mobility," *J. Field Robot.* **23**(1), 839–861 (2006).
11. J. Larson and M. Trivedi, "Lidar Based Off-Road Negative Obstacle Detection and Analysis," *Proceedings of the IEEE International Conference on Intelligent Transportation Systems* (2011).
12. J. Larson, M. Trivedi and M. Bruch, "Off-Road Terrain Traversability Analysis and Hazard Avoidance for UGVs," Technical Report (2010). Department of Electrical Engineering, University of California San Diego.
13. L. Matthies, A. Kelly, T. Litwin and G. Tharp, "Obstacle Detection for Unmanned Ground Vehicles: A Progress Report," **In**: *Proceedings of the IEEE Intelligent Vehicles Conference* (1995) pp. 66–71.
14. L. Matthies and A. Rankin, "Negative Obstacle Detection by Thermal Signature," *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2003).
15. T. Mioch, N. J. J. M. Smets and M. A. Neerincx, Assessing human-robot performances in complex situations with unit task tests, *Proceedings of the 21th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 621–626, Paris, France.
16. P. Papadakis and F. Pirri, "3D Mobility Learning and Regression of Articulated, Tracked Robotic Vehicles by Physics-Based Optimization," **In**: *Virtual Reality Interaction and Physical Simulation* (2012) pp. 147–156.
17. P. Papadakis, Content-Based 3D Model Retrieval Considering the User's Relevance Feedback. *PhD Thesis* (University of Athens, Athens Greece, 2009).
18. P. Papadakis, I. Pratikakis, S. Perantonis and T. Theoharis, "Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation," *Pattern Recognit.* **40**(9), 2437–2452 (2007).
19. M. Shevtsov, A. Soupikov and A. Kapustin, "Highly parallel fast KD-tree construction for interactive ray tracing of dynamic scenes," *Comput. Graph. Forum* **26**(3), 395–404 (2007).
20. S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Comput. Vis. Graph. Image Process.* **30**(1), 32–46 (1985).