2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 27 - October 1, 2021. Prague, Czech Republic

Using Depth Vision for Terrain Detection during Active Locomotion*

Ali H. A. Al-dabbagh and Renaud Ronsse

2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-1714-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/IROS51168.2021.9636077

Abstract— **Vision-based systems for terrain detection are ubiquitous in mobile robotics, while such systems recently emerged for locomotion assistance of disabled people. For instance, wearable devices embedding vision sensors can assist people in navigation; or guide lower-limb prosthesis or exoskeleton controller to retrieve gait patterns being adapted to the executed task (overground walking, stairs, slopes, etc.). In this research, we present a vision-based algorithm achieving the detection of flat ground, steps, and ramps, using a depth camera. The raw point cloud data obtained from the depth camera passes through multiple processing steps to extract environmental features, and then achieves terrain detection by feeding these features into a classification tree. The camera was mounted on a custom-made wearable tool that can be placed on the chest of human users. This contribution reports a pilot validation study with 6 healthy subjects moving in an indoor environment containing a rich set of different types of terrains. Our method can predict the locomotion modes up to three steps in front of the user. Moreover, it is able to perform terrain detection even if the path is partially occluded by another walker. The method accuracy with a cleared path was found to be above 90% for all locomotion modes.**

I. INTRODUCTION

The biomechanics of locomotion reveals that leg joints often produce mechanical energy in various tasks such as walking or stair ascending. Moreover, each of these tasks corresponds to different joint kinetic and kinematic patterns, even in steady-state. As consequence, active prostheses or exoskeletons are necessary to provide disabled users with the capacity to display healthy gait patterns. The control of such devices should be adaptive to the task being executed (walking, taking stairs, etc.) and several researchers recently developed such adaptive algorithms, for instance by using finite state controllers [1]. However, switching from one locomotion mode to another, requires a timely transition, sometimes even before the critical time or event. Interestingly, computer vision-based terrain detection systems can provide such predictive information about the surrounding environment for some steps in advance [2], [3]. More specifically, systems with depth sensing can accurately detect and locate environmental features in 3D spaces [4]. Therefore, the use of depth cameras in applications for locomotion assistance is increasing. This is due to the fact that such cameras are relatively cheep, compact, and can provide rich information about the environment [4].

*This work was supported by the Conseil de l'action internationale of UCLouvain and the European Community's H2020 Research and Innovation Programme under grant number 731931 (the CYBERLEGsPlusPlus collaborative project)

Ali H. A. Al-dabbagh and Renaud Ronsse are with Institute of Mechanics, Materials, and Civil Engineering; Institute of Neuroscience; and Louvain Bionics; Université catholique de Louvain, Louvain-la-Neuve, Belgium e-mails: ali.aldabbagh@uclouvain.be, renaud.ronsse@uclouvain.be

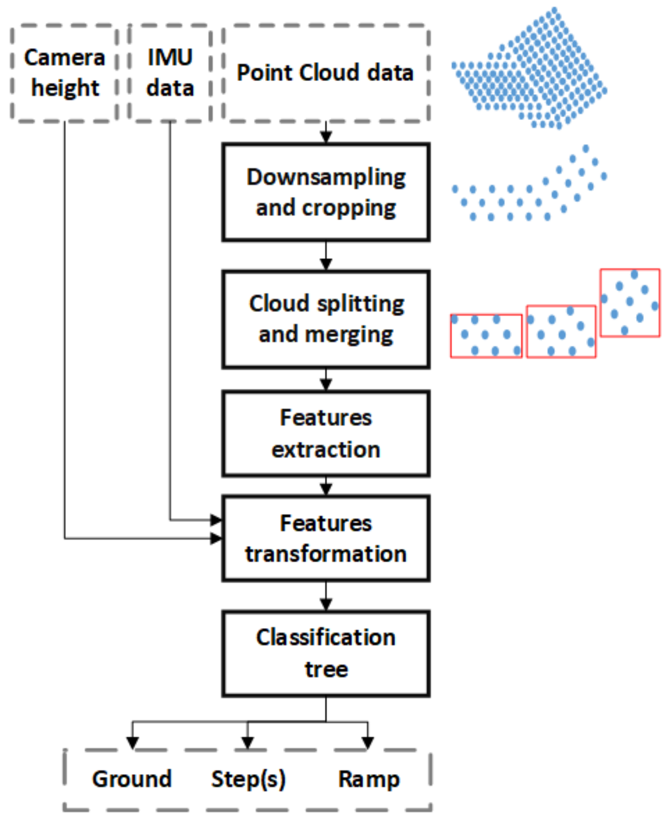

Fig. 1. Block diagram illustrating the overall terrain detection algorithm. Point cloud data and the camera orientation acquired from a depth camera and an IMU sensor, respectively, are the main inputs of the algorithm. After several processing steps, the algorithm estimates the current locomotion terrain as output, i.e. flat ground, stairs, and/or ramp

Several depth vision based terrain detection systems have been developed for human locomotion assistance [5]–[8]. These systems achieved remarkable accuracy in predicting the upcoming locomotion mode when they have been tested on paths which was free of subjects. However, it is important for such systems to work in environments being partially occluded by other people — potentially moving — since this is ubiquitous in urban contexts. Similarly, it is useful to predict the transition between two different types of terrains several steps in advance, in order to predict the number of steps to come before the transition, and thus to handle this transition in a smooth and predictive way. Accordingly, the contributions of this paper lie in the implementation of a depth vision based terrain recognition system with terrains probability estimation method predicting the locomotion modes up to three steps ahead of the user. Moreover, we conducted pilot experiments with another walker partly occluding the visual field in front of the user, in order to test the system robustness to cluttered environments. Also, our algorithm is computationally cheaper than recently published concurrent approaches, owing to the features transformation method we used.

The paper is organized as follows: Section II provides a description of the implemented algorithm, hardware used for validation, experimental protocols, and the accuracy estimation method. Section III reports and discusses the experimental results assessing the overall system performance, in particular regarding accuracy. Finally the paper ends with a conclusion.

II. Methods

We implemented a computer vision-based algorithm for terrain recognition using a depth camera. The algorithm has two main stages. Firstly, it processes the point cloud data to obtain the detected terrains for the whole detection range. Secondly, it splits the detection range into several regions which corresponds to the three upcoming steps to identify the putative locomotion mode for each of them. Accordingly, we will go through each stage in details in the next two sub-sections, while the rest of this section describes the system hardware and the experimental protocols assessing its performance.

A. Terrain detection algorithm

The raw point cloud data obtained from the depth camera passes through multiple processing steps to extract environmental features, and thus achieve terrain detection as shown in Fig.1. We will go through each step in following paragraphs.

1) Downsampling and Cropping: This step aims at reducing the computational load by minimizing the number of points in the cloud. Indeed, the high density of point cloud data causes processing delay, and not all the points bring a significant part of new information. The point cloud resolution is down sized by using algorithms 3D voxel grid filter, with a leaf size of 3cm [9]. To further reduce the computational cost, a region of interest (ROI) of one meter wide in front of the user was specified in order to eliminate data lying out of this region, similarly as in [10].

2) Cloud split and merge : This step is useful to identify the transition edges between grounds of different inclination, such as up/down ramps and flat ground. The point cloud is divided into three regions as a function of their closeness to the camera. About 700 points are thus kept for each region and the global normal vector of the plane containing these points is estimated. If the orientation difference between two adjacent regions is found to be less than 5 degree, then both groups of points are assumed to belong to the same plane and are thus merged. Otherwise, they are kept separated for the following computational steps. Consequently, this step outputs between one and three group(s) of points being passed to the next step.

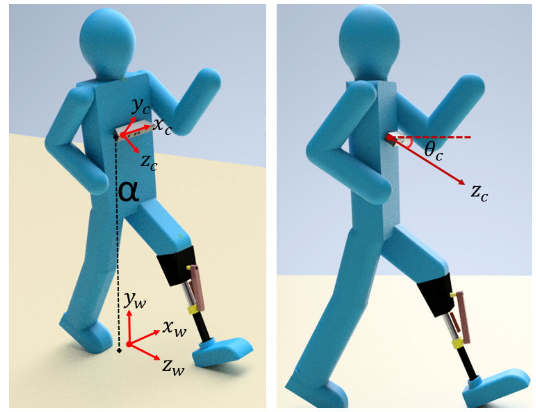

Fig. 2. The left panel shows the relationship between the camera frame (x_c, y_c, z_c) and the local world frame (x_w, y_w, z_w). The right panel shows the orientation of the camera frame with respect to the horizontal plane, in the sagittal plane

3) Feature extraction: In this step, we extract environmental features of each group of the point cloud from the previous step. This is achieved through the following sequence of processing:

- For each point, a normal vector is estimated from its neighboring points [11] based on Principal Component Analysis (PCA) [12]–[14]. This approach is efficient and has been used in other systems [15]–[17]. It computes the eigenvalue and eigenvector of a covariance matrix created from direct neighboring points. The normal direction is estimated as the eigenvector having the smallest eigenvalue [13].
- The point cloud data is then clustered into several groups using a regional growing algorithm. It starts from a point with minimum curvature called *seed*, and expands the region to include neighboring points having quasi-parallel normals and similar curvature [12]. When the region cannot be expanded anymore, a new seed is chosen, and the process is repeated until a pre-specified threshold is reached. This algorithm outputs a set of points (cluster) belonging to the same smooth surface. The regional growing algorithm has a good physical relevance since the planes are detected from a closed region depending on single element (seed) rather than from set of uncorrelated points distributed in the scene [12], [13].
- Finally, a plane is fitted over each cluster using the RANdom SAmple Consensus (RANSAC) algorithm developed by [18]. Then, for each fitted plane, its normal vector $\mathbf{n^c} = (n^c_x, n^c_y, n^c_z)^T$ and centroid $\mathbf{o^c} = (o^c_x, o^c_y, o^c_z)^T$ are computed and thus define the plane orientation and location in the 3D environment, with respect to the coordinate frame attached to the camera (x_c, y_c, z_c) (see Fig.2)

4) Features transformation: Point cloud data acquired from a depth sensor are provided in a coordinate frame being centered on the camera, and this causes issues for proper terrain detection. For instance, if the camera points downwards, level ground will be detected as a ramp [6]. A common solution is to transform the reference coordinate system to the ground or local world coordinates based on the measured instantaneous orientation of the camera. This orientation can be provided e.g. by an IMU that is attached to the camera. Typically, this transformation is applied on the whole point cloud just after acquisition, and thus before feature extraction [6], [7], [13]. Here, we applied the transformation on the obtained plane features (centroid and slope), since it is not necessary to rotate and translate all points of the acquired cloud. We observed that this led to save about 10% of computational time as compared to the former approach consisting in transforming the whole cloud. Fig.2 defines the coordinate frame attached to the camera (x_c, y_c, z_c) and the so-called "local world frame" (x_w, y_w, z_w) that is used for terrain recognition. This local world frame is moving with the user but is kept with y_w pointing upwards and z_w pointing in the user's direction of locomotion. Assuming that the camera is only translating and rotating in the $x_w - z_w$ plane, the relationship between both frames is thus given by the homogeneous transformation matrix:

$$\mathbf{H_c^w} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_c & -\sin\theta_c & \alpha \\ 0 & \sin\theta_c & \cos\theta_c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where θ_c is the camera rotation angle and α is the distance between the ground and the camera, as represented in Fig.2. Accordingly, the plane orientation and centroid can be transformed back to the local world frame by pre-multiplying their coordinate by the transformation matrix, i.e. $(n^w, 1)^T = \mathbf{H_c^w}(n^c, 1)^T$ and $(o^w, 1)^T = \mathbf{H_c^w}(o^c, 1)^T$.

In order to make the system robust to the intrinsic sensor noise of the IMU, θ_c was kept constant and equal to a value calibrated when the subject was standing still, , if the detected locomotion mode was ground or up-/down-ramps. In both stairs modes, θ_c was extracted in real-time from IMU data. The number of detected planes was triggering the system to switch between fixed and instantaneous IMU data. In all cases, α was calibrated by measuring the distance between the camera and the ground once the device was mounted on a given subject.

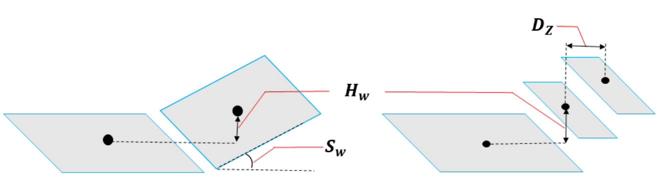

Fig. 3. Features fed to the classification tree to achieve various terrain recognition. S_w is the plane orientation angle, H_w is the plane height with respect to the user's feet and D_z is the relative distance between two consecutive planes in the forward direction

5) Classification tree: The following features are extracted for each plane whose coordinates have been transformed in the local world frame during the previous step:

- S_w=atan(n_y^w/n_z^w) is the plane orientation angle, defined as the angle between the plane normal vector and the vertical axis y_w.
- $H_w = o_y^w$ is the coordinate in the plane centroid in the vertical direction, i.e. the plane height with respect to the user's feet.
- If two different adjacent planes are detected, their relative distance in the forward direction is computed as $D_z = |o_{z1}^w - o_{z2}^w|$.

This extra feature will turn out to be useful for improving the detection accuracy of staircases by preventing the system from getting boxes, tables, flat obstacles, and low leveled shelves on the ground mixed up with stair steps. The features passed to the classification tree are pictured in Fig.3. The tree applies several tests on these features which leads to various terrain predictions as shown in Fig.4. It starts by examining the features against ramp, stair, and finally flat ground candidates. For instance, the detected plane is classified as being a ramp up if 5°< S_w <25° and 0.1m< H_w <1m. If this up-ramp candidacy test is not positive, then the plane is examined against down-ramp conditions, etc. Planes having feature values lying outside the conditions of ramp, stairs, and flat ground terrains are considered as obstacles and labeled as "No Detection" (ND) by the algorithm. A specific color is assigned to each detected type of terrains for visualization purposes.

B. Predicting locomotion terrain probability several steps ahead

A typical depth camera can record environmental information for more than one meter of distance. Therefore, it is possible to exploit such information to estimate the locomotion terrain for several steps in advance. Accordingly, the classified planes were segregated into three regions approximately corresponding to the three upcoming steps in front of the user. More precisely, each step was taken with a stride equal to 60 cm for ground and ramp and 30 cm for staircase. For each step, a probability of walking on a given type of terrain was computed in the following way:

1) Each point belonging to the down-sampled cloud was assigned with a color corresponding to the color of the detected terrain of the corresponding plane (see Fig.4).
2) The probability of walking on a given terrain for a given step was computed as the fraction of points of the corresponding color in the corresponding step region, over the total amount of points in that region.

As output, the algorithm thus provides the probability of walking on each possible terrain for three steps upfront of the user. In order to get the most probable terrain, a maximum filter is applied on these probabilities, and this discrete single estimate is finally low-pass filtered by a majority vote over the last 3 samples. This final discrete estimate is stored in time-dependent vectors $T_{i,detected}$, sampled at the same frequency as the point cloud recording, where $i \in 1,3$ corresponds to the three steps upfront of the user.

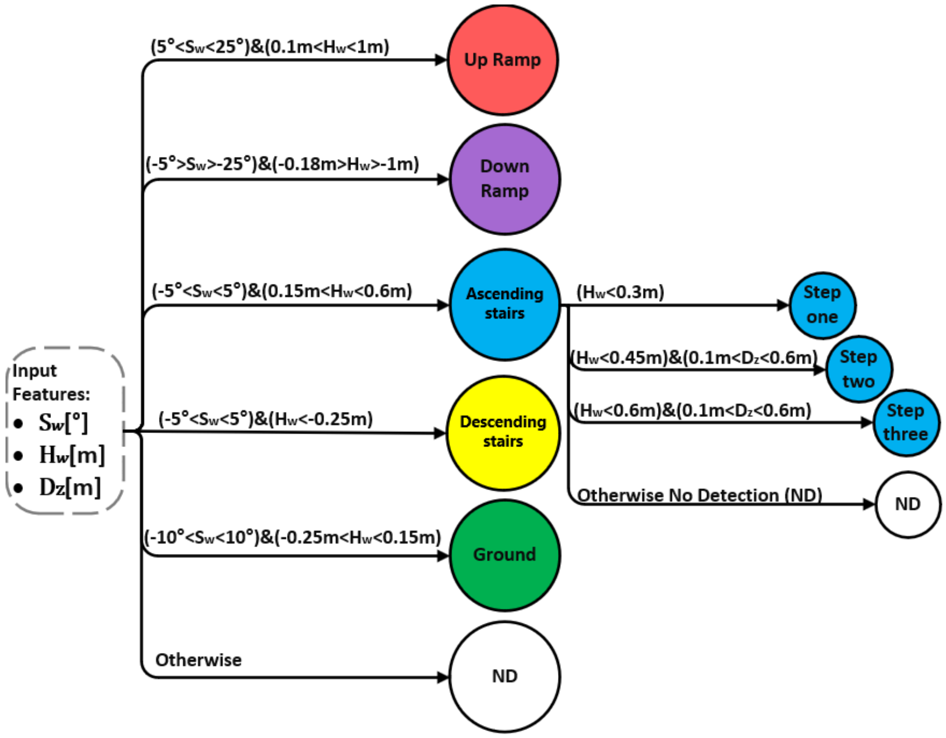

Fig. 4. The classification tree examines the extracted features S_w, H_w, and optionally D_z, against several conditions to achieve terrain identification. First, it begins by testing the features for the ramp conditions, then for the stair conditions, and finally for the ground conditions. The plane is considered to be an obstacle if its features lie outside all the above constraints. Colors given for each type of detected terrain correspond to the colors assigned to the point cloud of raw data for visualization purposed (see e.g. Fig. 8). If ascending stairs are detected, a further step is applied using the third feature D_z to confirm the validity of the steps being successively detected. For instance, the second step is considered as being an obstacle (No detection) unless the first one is properly detected and similarly for the third step (see the text for more details).

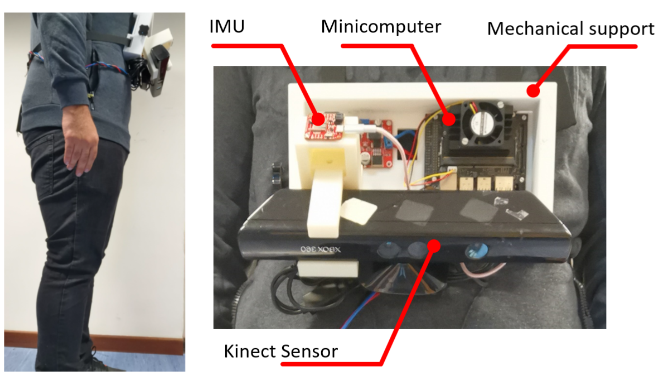

Fig. 5. The designed wearable tool to carry the system hardware consisting of a depth camera, a mini computer, and an IMU

C. Hardware

The algorithm was implemented in the Robot Operating System (ROS), and using pre-existing Point Cloud Library (PCL) [9]. A wearable tool was designed for the purpose of data collection (see Fig.5). The tool consists of an IMU (9DoF Razor-SparkFun), a Kinect camera (Xbox 360), and minicomputer (Jetson Nano). This minicomputer performs the first step of the algorithm (downsampling and cropping) and records downsampled point clouds, raw IMU data, and RGB data being used for off-line labeling of the ground truth of the detected terrain. The data are recorded with different sampling rates (5fps for the point cloud data, 200 fps for IMU data and 30fps for the RGB data). The wearable tool is designed to be mounted on the subject chest with belts. This sensor configuration tends to provide more stable images than other configurations such as on the leg or head [12], [13].

D. Experimental methods

An indoor path on the UCLouvain campus of Louvain-la-Neuve was selected to assess the system performance (see Fig.6). The path contains different kind of targeted terrains such as ground, stairs, and ramps. The system was tested in the two following conditions:

- Uncluttered environment: the system was tested with six healthy subjects (4 males and 2 females, between the ages of 27 and 32). Each individual wore the device and walked through the specified path while it was free of other people, at their preferred walking speed.
- Cluttered environment: in order to reproduce a real life scenario in which the walking path is not always free of other people, we further conducted pilot tests to assess the detection robustness with partially occluded footages. This is important since real urban environments cloud be cluttered with other people or obstacles. Accordingly, two healthy subjects participated to this pilot experiment. The first subject played the user role and was thus wearing the device, and the other subject

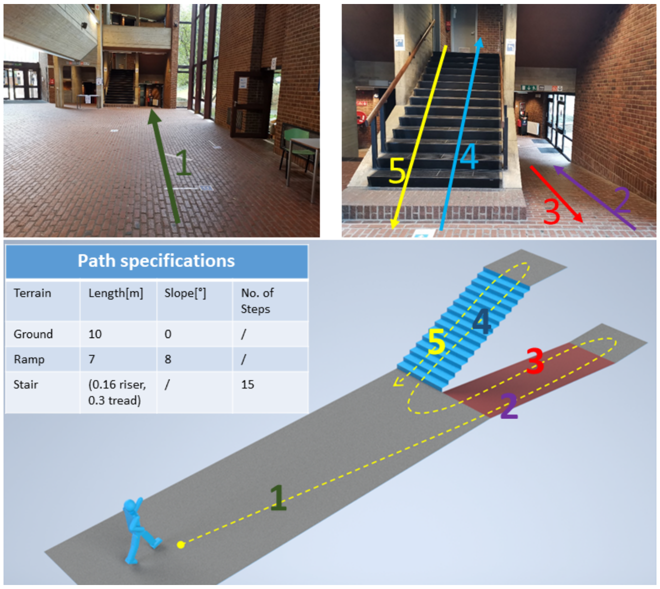

Fig. 6. Top pictures are snapshots of the real experimental environment. Each subject walked on the path with the wearable tool starting from 1 (flat ground) and finishing with 5 (downstairs). The bottom panel shows a 3D representation of the path with its specifications

played the role of the impediment and was thus walking in front of the user on the same path. This experimental condition was conducted twice. In the first trial, the leader walked very close upfront of the user (approximately 0.8 to 1.2 m). In the second trial, the leader walked a bit further away from the user (approximately 1.2 to 2 m).

E. Accuracy estimation

In order to estimate the system accuracy in detecting each locomotion mode in uncluttered environment, the recorded data with the 6 subjects were categorized based on the five locomotion tasks identified above and labeled as ground (G), down-ramp (DR), up-ramp (UR), ascending stairs (AS) and descending stairs (DS). Ground truth labeling was performed by an expert (first author of this paper), using the RGB footages recorded during the experiments. A typical evolution of T_{actual} vs. $T_{1,detect}$ for the full path (including all locomotion tasks) is pictured in Fig.7. Computing the percentage of samples where $T_{1,detect}$ and T_{actual} are equal over the experimental trial gives an overall estimate of the system accuracy for each of the three steps ahead of the user, independently. Similarly, confusion matrices between each of the conditions can be computed by reporting the fraction of samples where both vectors disagree.

III. Results and Discussion

A. Uncluttered environment

Typical detection samples for different kind of terrains are provided in Fig.8 , with the color code corresponding to the one of Fig.4. It can be seen that the system provided a good separation of ground planes at the transition between

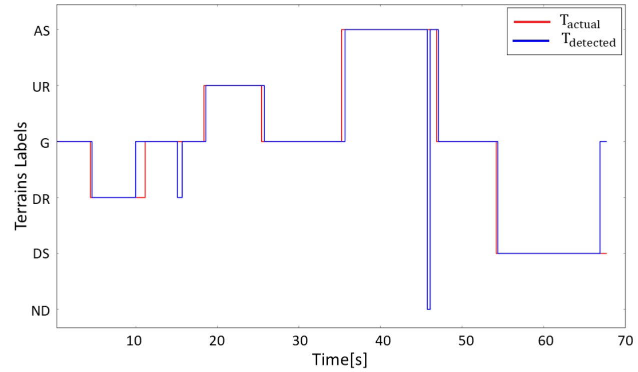

Fig. 7. Actual (T_{actual}) vs. detected (T_{detect}) terrains recording in a typical trial involving ground (G), down-ramp(DR), up-ramp(UR), ascending stairs(DS) and descending stairs(DS).

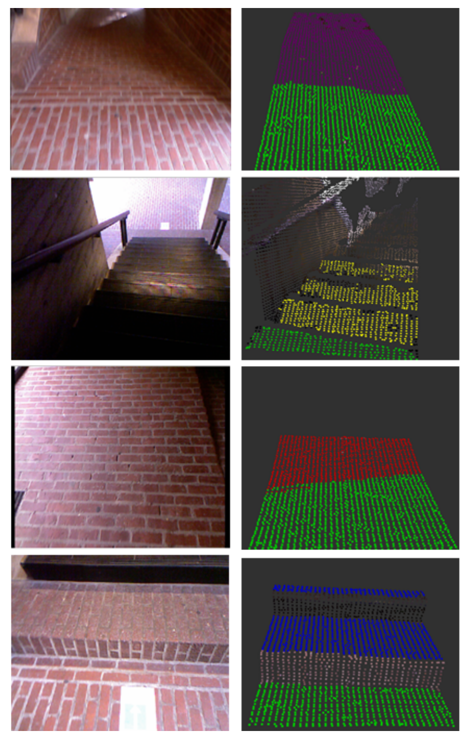

Fig. 8. Detection samples of different terrains. The left column shows real RGB footages while the right column shows the algorithm recognition results. Green points refers to ground detection, while red and purple corresponding to up-ramp and down-ramp respectively. Steps of ascending and descending stairs are pictured in blue and yellow points respectively.

different kinds of terrains. In particular, the separation between the ground and ramp was well detected although there was no physical separation between them. Such accurate separation holds promise if fed to the controller of an active assistive device for locomotion (such as an exoskeleton or a prosthesis) since the new locomotion mode should be triggered at the right time. The average system accuracy was computed for each locomotion mode and reported in Fig.9 for one to three steps ahead. To have a better insight about the wrong detections, we computed the confusion matrix for the three steps, and they are provided in Tables I to III. For the first two steps, the system achieved above 90% of detection

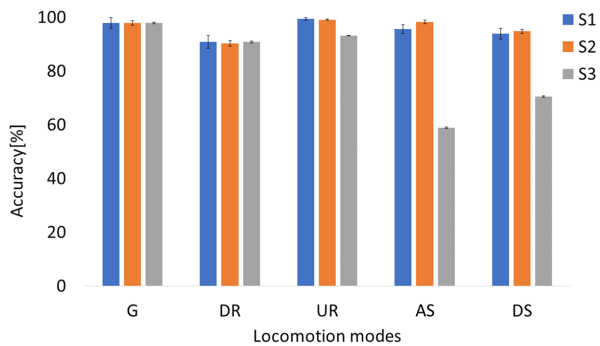

Fig. 9. Average accuracy for each locomotion task and for three steps in front of the user in uncluttered environment. S1, S2 and S3 refer to steps 1 to 3. Whiskers represent the standard error of the mean, obtained across the 6 different users. Labels correspond to G: flat ground; DR: down-ramp; UR: up-ramp; DS: descending stairs; and AS: ascending stairs.

accuracy for all kind of terrains. However, the system reached lower accuracy for the third step, corresponding to the one being the furthest away from the user. In particular, the detection accuracy got degraded for detecting ascending and descending staircases. This was mainly due to the fact that in this case, the sensor was lacking a complete vision of the whole staircase, since the edge of the third step was not detected most of the time. The ND (no detection) label corresponded indeed to the most frequent confusing detection in these cases. A possible solution would be to tilt the sensor up a bit more in order to have a full vision of the third step, but this would deteriorate the capacity to detect the first step since the terrain just in front of the user would be out of the camera field of view.

B. Cluttered environment

To the best of our knowledge, testing vision-based locomotion detection on terrains partially occluded by other people has never been addressed so far. However, it is a matter of fact that typical urban environments are crowded most of the time, and that people tend to like walking in a group. Accordingly, we assessed preliminary performance of our system while the path was partially occluded by another person, walking at two different distances upfront. Detection estimates corresponding to various locomotion modes are displayed in Fig.10. The accuracy was calculated for both trials, and again for three steps upfront of the user, as illustrated in Fig. 11 and 12. When the leader was close to the user (0.8-1.2m distance between them), the system achieved lower accuracy in comparison to the results reported for a clear path, mainly for the third step upfront. This is likely because the leader occluded any relevant visual information for this third step. In the second trial, where the leader walked further away from the user (1.2-2m distance between them), we obtained similar results for the first two steps as with clear path, and again a slightly lower accuracy for the third step.

TABLE I

CONFUSION MATRIX FOR STEP ONE. (ND) REFERS TO NO DETECTION.

	G	DR	UR	AS	DS	ND
G	98.14	0	1.85	0	0	0
DR	9.54	90.45	0	0	0	0
UR	0	0	99.5	0.5	0	0
AS	2.62	0	0.32	95.08	0	1.96
DS	6	0.8	0	0	93.14	0

TABLE II

CONFUSION MATRIX FOR STEP TWO

	G	DR	UR	AS	DS	ND
G	98.14	0	1.85	0	0	0
DR	8.04	90.45	0	0	1.5	0
UR	0.52	0	98.95	0.52	0	0
AS	1.02	0	0.68	98.29	0	0
DS	4.45	1.61	0	0	93.92	0

TABLE III

CONFUSION MATRIX FOR STEP THREE

	G	DR	UR	AS	DS	ND
G	98.14	0	1.85	0	0	0
DR	9.14	90.85	0	0	0	0
UR	8.612	0	91.38	0	0	0
AS	0.36	0	4.39	59.34	0	35.89
DS	1.62	6.91	0	0	69.91	21.54

C. Discussion

The system we implemented showed remarkable detection results in both uncluttered and cluttered environments. Our algorithm is built upon a framework that has already been tested by others [4], although it requires a priori less computational resources because we transformed only the detected plane features into the local world reference frame, rather than the whole point cloud. Consequently, this approach is promising for wearable systems with limited processing capabilities. The other major contribution of our study is to outline a framework that can provide a probabilistic estimate of the terrain to be traversed up to three steps ahead of the user. This predictive capacity is likely to be critical if our system is coupled to a locomotion assistive device, it order to trigger the transition for one locomotion mode to another in a smooth and timely fashion. A possible extension would be to adapt the size of the locomotion stride as a function of the actual one of the user.

However, the current implementation of our system has some limitations. For instance, no real-time camera orientation was used in the flat ground and ramp modes. This means that large changes of the camera orientation during locomotion can lead to wrong detection in these modes. Also, due to hardware limitations of the selected camera, the system cannot work under the direct sunlight, so we could not test our algorithm in outdoor environments. Accordingly, we are working on developing an improved version of the system with a more compact sensor which works in indoor and outdoor environments. Next, the classification methods that was used uses hand-tuned thresholds that are based on geometric features of typical environments. Although

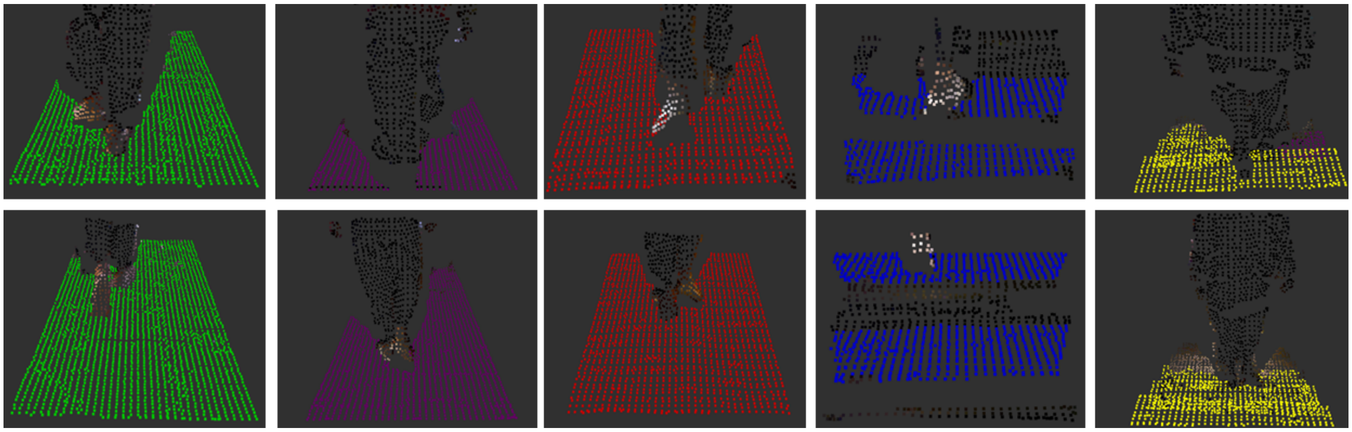

Fig. 10. Detection results during the five locomotion tasks while the terrains were partially occluded by another subject. The first row of images show the detection results while the leader subject was 0.8-1.2m upfront of the user (first trial). The second row shows detection results while the leader was 1.2-2m upfront of the user (trial 2). The color code is the same as for Fig.8.

this provides a direct mapping between the environment geometry and the resulting classification, this might lead to poor generalization to other paths with different features. We are therefore working on the development of a data-driven classification method using machine learning approaches instead of fixed classification thresholds. Finally, more experiments should be conducted with several paths, in order to precisely assess the system performance and generalizability.

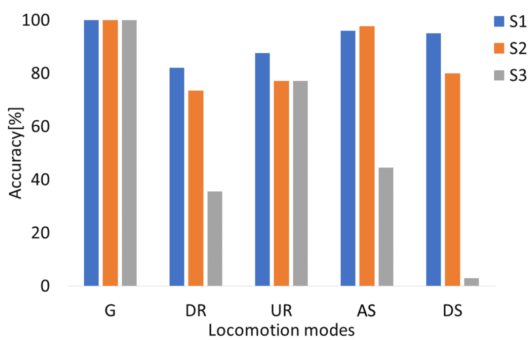

Fig. 11. Accuracy for each locomotion task and for three steps in front of the user in cluttered environment when the leader walked 0.8-1.2m upfront the user (1st trial). S1, S2 and S3 refer to steps 1 to 3. Labels correspond to G: flat ground; DR: down-ramp; UR: up-ramp; DS: descending stairs; and AS: ascending stairs.

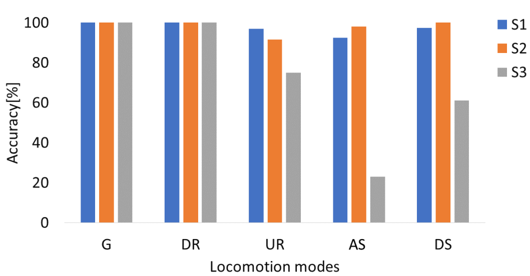

Fig. 12. Accuracy for each locomotion task and for three steps in front of the user in cluttered environment when the leader walked 1.2-2m upfront of the user (2nd trial). S1, S2 and S3 refer to steps 1 to 3. Labels correspond to G: flat ground; DR: down-ramp; UR: up-ramp; DS: descending stairs; and AS: ascending stairs.

IV. Conclusion

In conclusion, we developed a computer vision based algorithm which can provide a disabled user, a prosthesis or exoskeleton controller with information about the type of locomotion terrain three steps forward, with remarkable accuracy. This can contribute in restoring navigation skills to someone e.g. with vision disorders, or to improve planning capacities of walking assistance devices. Although the system has some limitations, pilot results showed that our algorithm is robust to partial occlusions due to another human walking upfront of the user at several distances, at least for the first two prediction steps. However, more experiments has to be conducted with several paths and more subjects to assess the system actual performance more precisely. Therefore, we are currently running further validation tests of this algorithm, with a more compact sensor and in both indoor and outdoor environments. Future work will also consist in refining the classifier with methods from machine learning.

References

[1] M. R. Tucker, J. Olivier, A. Pagel, H. Bleuler, M. Bouri, O. Lambercy, J. del R Millán, R. Riener, H. Vallery, and R. Gassert, "Control strategies for active lower extremity prosthetics and orthotics: a review," *Journal of neuroengineering and rehabilitation*, vol. 12, no. 1, p. 1, 2015.

[2] B. Laschowski, W. McNally, A. Wong, and J. McPhee, "Comparative analysis of environment recognition systems for control of lower-limb exoskeletons and prostheses," in *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2020, pp. 581–586.

[3] N. E. Krausz and L. J. Hargrove, "A survey of teleceptive sensing for wearable assistive robotic devices," *Sensors*, vol. 19, no. 23, p. 5238, 2019.

[4] A. H. Al-dabbagh and R. Ronsse, "A review of terrain detection systems for applications in locomotion assistance," *Robotics and Autonomous Systems*, p. 103628, 2020.

[5] K. Zhang, J. Luo, W. Xiao, W. Zhang, H. Liu, J. Zhu, Z. Lu, Y. Rong, C. W. de Silva, and C. Fu, "A subvision system for enhancing the environmental adaptability of the powered transfemoral prosthesis," *IEEE Transactions on Cybernetics*, 2020.

[6] K. Zhang, C. Xiong, W. Zhang, H. Liu, D. Lai, Y. Rong, and C. Fu, "Environmental Features Recognition for Lower Limb Prostheses Toward Predictive Walking," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 3, pp. 465–476, 2019.
[7] N. E. Krausz, T. Lenzi, and L. J. Hargrove, "Depth sensing for improved control of lower limb prostheses," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 11, pp. 2576–2587, 2015.
[8] B. Berenguel-Baeta, M. Guerrero-Viu, A. Nova, J. Bermudez-Cameo, A. Perez-Yus, and J. J. Guerrero, "Floor extraction and door detection for visually impaired guidance," in *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2020, pp. 1222–1229.
[9] R. B. Rusu and S. Cousins, "Point cloud library (pcl)," in *2011 IEEE international conference on robotics and automation*, 2011, pp. 1–4.
[10] H.-H. Pham, T.-L. Le, and N. Vuillerme, "Real-time obstacle detection system in indoor environment for the visually impaired using microsoft kinect sensor," *Journal of Sensors*, vol. 2016, 2016.
[11] A. Ciobanu, A. Morar, F. Moldoveanu, L. Petrescu, O. Ferche, and A. Moldoveanu, "Real-time indoor staircase detection on mobile devices," in *2017 21st International Conference on Control Systems and Computer Science (CSCS)*. IEEE, 2017, pp. 287–293.
[12] A. Pérez-Yus, G. López-Nicolás, and J. J. Guerrero, "Detection and modelling of staircases using a wearable depth sensor," in *European Conference on Computer Vision*. Springer, 2014, pp. 449–463.
[13] A. Perez-Yus, D. Gutiérrez-Gómez, G. Lopez-Nicolas, and J. Guerrero, "Stairs detection with odometry-aided traversal from a wearable rgb-d camera," *Computer Vision and Image Understanding*, vol. 154, pp. 192–205, 2017.
[14] K. Yang, K. Wang, R. Cheng, and X. Zhu, "A new approach of point cloud processing and scene segmentation for guiding the visually impaired," 2015.
[15] M. Bellone, G. Reina, N. I. Giannoccaro, and L. Spedicato, "Unevenness point descriptor for terrain analysis in mobile robot applications," *International Journal of Advanced Robotic Systems*, vol. 10, no. 7, p. 284, 2013.
[16] M. Bellone, A. Messina, and G. Reina, "A new approach for terrain analysis in mobile robot applications," in *2013 IEEE International Conference on Mechatronics (ICM)*. IEEE, 2013, pp. 225–230.
[17] G. Reina, M. Bellone, L. Spedicato, and N. I. Giannoccaro, "3d traversability awareness for rough terrain mobile robots," *Sensor Review*, vol. 34, no. 2, pp. 220–232, 2014.
[18] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.