Self-Supervised Learning of Depth-Based Navigation Affordances from Haptic Cues

José Baleia[1], Pedro Santana[2,3] and José Barata[1]
[1] CTS-UNINOVA, Universidade Nova de Lisboa, Portugal
[2] ISCTE - Instituto Universitário de Lisboa, Portugal
[3] Instituto de Telecomunicacões, Portugal
Emails: jrfbal@hotmail.com, pedro.santana@iscte.pt, jab@uninova.pt

***Abstract*—This paper presents a ground vehicle capable of exploiting haptic cues to learn navigation affordances from depth cues. A simple pan-tilt telescopic antenna and a Kinect sensor, both fitted to the robot's body frame, provide the required haptic and depth sensory feedback, respectively. With the antenna, the robot determines whether an object is traversable by the robot. Then, the interaction outcome is associated to the object's depth-based descriptor. Later on, the robot to predict if a newly observed object is traversable just by inspecting its depth-based appearance uses this acquired knowledge. A set of field trials show the ability of the to robot progressively learn which elements of the environment are traversable.**

***Keywords*—autonomous robots, self-supervised learning, affordances, terrain assessment, depth sensing, robotic antenna**

I. Introduction

Since the first invertebrate ventured out of the Panthalassa, the ability to navigate through the environment became crucial for the species survival. Nature evolved in order to allow for different means of interaction and learning mechanisms, some insects developed antennas to sense the surroundings, while mammal's brains grew to support the big influx of information provided by non-haptic feedback, like hearing, vision, and olfactory perception.

Inspired by Nature, in which visual and haptic sensory feedback are known to be jointly exploited in the Human brain [1] this paper presents a haptic robot-environment interaction system for self-supervised learning of vision skills for safe navigation. For this purpose, the robot is provided with a mechanism to learn a mapping between the volumetric appearance of obstacles, given sensory data provided by a depth sensor, and their bendability, perceived by physically interacting with them with a small antenna (see Fig. 1). As interactions unfold, the robot grows its ability to properly assess the cost of navigating the environment from its depth sensor and, consequently, reducing the need for physical interactions. As a consequence, the robot's spatial reasoning look-ahead grows significantly, which is key to ensure a safe navigation.

From an engineering standpoint, this incremental learning approach reduces considerably the design space. The robot designer needs only to specify that the inability of the antenna to reach a given point in space due to obstruction is a sufficient cue that the object is not bendable and, thus, not traversable by the robot. Conversely, designing a similar classifier but operating directly on depth data would be a much more complex endeavor.

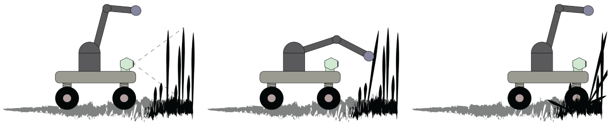


Fig. 1. Proposed system's major steps. (Left) The robot finding an object with its depth sensor. (Middle) As the object's class is still new to the robot, the latter physically interacts with so as to learn its traversability. (Right) The robot overcoming the traversable object.

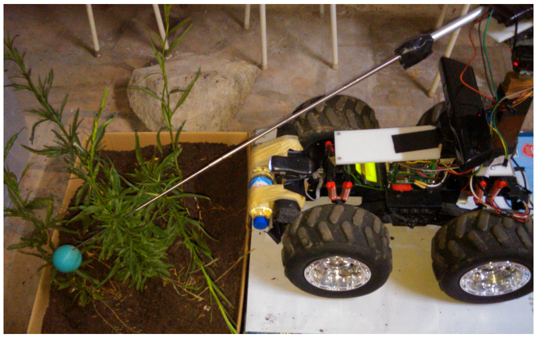


Fig. 2. The robot prototype with its antenna stretched to its maximum range.

Having the handcrafted haptic classifier supervising the automatic learning of the depth-based classifier provides vision-based far field terrain assessment without growing the engineering design space. The incremental learning about the bendability of the objects occur in a self-supervised way, in the sense that the labels associated to each training data are generated by the robot itself via haptics-based interactions. In a way, the robot is learning what the objects present in the environment afford in terms of successful motor actions, i.e., can the object be overcome or not. This resembles the affordance principle studied by Gibson for the animal kingdom [2]. The concept of affordances link the ability of a subject through its actions to the features of the environment and, so, to learn an affordance the agent needs to interact with the environment.

The affordances concept has been exploited with antenna-based [3] and whiskers-based [4], [5] interactions for the case of object manipulation and recognition. The application to navigation cost, the core of this paper, has already been reported as well [6]–[8]. However, this previous work considers learning affordances from full-body interactions, i.e., by moving the robot against the objects. Conversely, this paper proposes assessing navigation cost with a robotic antenna, which reduces the robot's risk of the getting stuck or damaged as well as it allows a finer analysis of the object. To assess the proposed system, a robotic prototype was developed (see Fig. 2). The prototype is based on a $45\,\mathrm{cm} \times 35\,\mathrm{cm} \times 65\,\mathrm{cm}$ 4-wheeled robot with differential locomotion, fitted with a custom telescopic antenna with pan tilt control. The antenna is capable of stretching up to 1 m and its pan and tilt cover 180 in both axis, respectively. As depth sensor, the robot uses a Microsoft Kinect, which employs modulated light to capture tridimensional point clouds of the environment. This sensor has been shown to be able to acquire accurate enough 3D information for vegetation classification [9]. It is applicable robustly outdoors during the night and at most in the presence of dim daylight. For daylight operation the robot would have to be equipped, for instance, with a binocular vision sensor. As the noise model of these two sensory modalities is rather similar, the proposed model should be easily applicable to binocular vision and, as a result, enable daytime outdoors operation. The system is implemented on top of Robotics Operating System (ROS) [10] and relies on Point Cloud Library (PCL) [11] for low-level point clouds processing.

This paper is organized as follows. Section II describes the proposed system. Then, the results obtained from a set of field trials are presented in Section III. Finally, a set of conclusions and future work avenues are given in Section IV.

II. The Proposed System

A. System Workflow

This section describes the proposed system. The system aims at incrementally developing the ability to assess the cost of navigating in natural ground environments. For this purpose the robot learns a mapping between the appearance of obstacles, given sensory data provided by a depth sensor, and their bendability, perceived by physically interacting with them with a small antenna.

Fig. 3 shows an overview of the system's workflow. While executing a given mission, e.g., moving towards a specified waypoint, the robot may face an object. This object can be traversable (e.g., vegetation) or not (e.g., a rock). To assess it, the robot creates a 3D descriptor of the found object and uses it to search its memory for the outcome of previous encounters with similar objects. If these previous encounters taught the robot that the object is traversable then the robot does not expend the effort of avoiding it. However, while traversing the object the robot may find itself stuck and, consequently, needs to update the memory to report that the object is not traversable. The outcome of consulting the memory may produce a low confidence result when the object is being seen for the first time or there have been ambiguous previous interactions with it. In this case the robot opts to perform an haptic interaction with the object. The higher the confidence the robot is on the contents of the memory, the coarser the haptic interaction must be. This allows the robot to reduce the time of interaction as the object gets known and, in the limit, when confidences rises to a certain level the interaction is skipped altogether, The result of the interaction is then used to update the memory in terms of how traversable is the object.

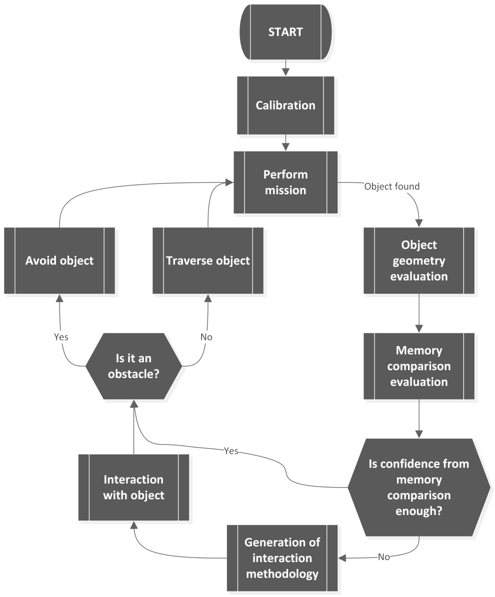

START
Calibration
Perform mission
Object found
Object geometry evaluation
Memory comparison evaluation
Is confidence from memory comparison enough?
Yes
No
Generation of interaction methodology
Interaction with object
Is it an obstacle?
Yes
No
Avoid object
Traverse object

Fig. 3. Proposed system's workflow.

B. Haptic-Visual Mapping

With only 3 degrees of freedom, the telescopic pan-tilt controlled antenna's inverse-kinematics can be sufficiently approximated with a simple tridimensional transformation from cartesian to polar coordinates. This transformation allows the antenna to reach a given tridimensional point in its workspace. These points are those determined as interesting in the tridimensional point cloud extracted from the depth sensor. A homogeneous point $\mathbf{p} = [x\,y\,z\,1]^T$ in the robot's workspace is described with respect to the antenna and sensor frames of reference as $\mathbf{p}_\mathcal{S} = [x_\mathcal{S}\,y_\mathcal{S}\,z_\mathcal{S}\,1]^T$ and $\mathbf{p}_\mathcal{A} = [x_\mathcal{A}\,y_\mathcal{A}\,z_\mathcal{A}\,1]^T$, respectively. Given the point coordinates in the camera frame of reference, the robot uses a 4×4 rigid body transformation matrix $\mathbf{M}$ to get the corresponding coordinates in the antenna's frame of reference, $\mathbf{p}_\mathcal{A} = \mathbf{M}\mathbf{p}_\mathcal{S}$.

To learn matrix $\mathbf{M}$, the robot antenna performs a babbling behaviour in order to cover its configuration space. Simultaneously, the robot tracks the antenna's end effector with the depth sensor. This allows the robot to accumulate a set of n correspondences between points in the antenna's and in the sensor's frames of reference, $\mathbf{p}^j_\mathcal{A} \leftrightarrow \mathbf{p}^j_\mathcal{S}, \forall j = 1, 2, \ldots, n$. Matrix $\mathbf{M}$ is estimated with a least-square SVD-based closed-form solution to the problem $\sum_{j=1}^{n} ||\mathbf{p}^j_\mathcal{A} - \mathbf{M}\mathbf{p}^j_\mathcal{S}||^2$ [12].

Before tracking the end effector, which takes a spheroid shape, the background is learned. This is done by storing an octree representation of the environment within the antenna's reach. Then, changes in the octree structure occurring while the antenna moves in the environment are taken as representative of foreground points. These correspond to the antenna and potential background moving objects. To reject false positives, a RANSAC [13] procedure is applied to all foreground points so as to extract the pose of the antenna's spheroid end-effector.

C. Object Descriptor

The robot is equipped with a depth sensor that produces a 3D point cloud of its surroundings. To ensure fast processing,

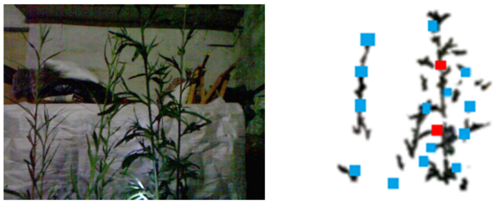


Fig. 4. Typical interaction points suggested by the system as motion plan. Red and blue filled squares represent the higher and lower scored points for the robot to interact, respectively.

the point cloud is down-sampled so as to ensure that no point is within a $1\,\text{cm}$ radius of another point. Then, the object is segmented from the background by simply dropping all the 3D points that are out of the antenna's reach. Finally, a descriptor of the object is built. The object's descriptor will represent the object in memory and will be used for comparisons with other objects. It must be rich enough for a robust comparison but simple enough for fast computations. Bearing this in mind, a set of four simple metrics based on two bidimensional histograms built from the 3D points distribution are considered.

Let us assume that the optical axis of the depth sensor is aligned with the robot's forward motion, i.e., parallel to the ground plane. The z-axis of the sensor is aligned with its optical axis, the y-axis is perpendicular to the ground plane pointing downwards, and the x-axis pointing to the right of the robot. Consider that all 3D points are projected to a bidimensional histogram coinciding with the xy-plane. A bin in this histogram represents the number of 3D points that are encompassed by the parallelepiped that crosses the corresponding small squared region of the xy-plane and extends to the sensor's maximum range. In the current implementation the histogram is composed of 16 x 14 bins.

Let us call *line* to the set of histogram's bins sharing the same y-coordinate. Let us call *cluster* to a set of adjacent occupied bins, in a given line, that are separate from other clusters by empty bins. The number of clusters found in line l is n_l. The number of bins corresponding to the largest cluster found in l is l_l. The density of points in a line l is d_l. Density is computed as the number of points in a line divided by the number of bins composing the line. The object's descriptor is a j-dimensional vector composed by these four metrics as computed for the h_x lines, $j = 4hx$.

D. Memory Recall

The memory is composed of descriptor-traversability tuples. A tuple is built by associating the descriptor of the observed object and the physical interaction binary-valued outcome. In the current implementation, forgetting has not been implemented. Therefore, all interactions are stored and maintained throughout the robot's lifecycle.

When facing an object, the robot will search for similar objects stored in memory in order to determine the most likely navigation cost of the object. The recall is done with a simple k nearest neighbour approach based on a similarity score computed between the descriptor of the observed object and the descriptors of the objects stored in memory. The similarity score is computed with the L1-distance, divided by the number of lines. To provide different relevance to each dimension of these vectors, they are weighted according to a set of empirically defined scalars prior to the distance calculation. These scalars sum up to the unity and, ideally, would be learned from data.

To determine whether the observed object is bendable or not, similarity scores of the neighbours that are traversable and that are not traversable are accumulated separately. Then, the final classification is the one associated to the highest accumulated score. The accumulated score associated to the obtained classification is taken as its confidence level. This confidence level serves the purpose of deciding whether the robot should traverse/avoid the object according to the memory recall output or it should physically interact with the object in order to improve its knowledge about it.

E. Haptic Interaction Motion Planning

The interaction between the robotic antenna and the object under assessment should be as efficient as possible, otherwise haptic interactions become time and energy over consuming. That is to say that the antenna's motion plan should be short yet still capable of allowing the robot to assess whether the object is bendable or not. The antenna's motion plan is defined by a set of 3D positions that must be reached in sequence. The order of the sequence is defined by the likelihood the robot has of assessing the bendability of the object when reaching the positions in question. Reaching the first position in the sequence has the highest chances of blocking the antenna's motion when the object is not bendable. If the motion gets blocked, then the robot does not need to check the next position in the sequence to reach a conclusion.

The position of each 3D point present in the point cloud is a candidate for the motion plan. However, as most 3D points are redundant in terms of interaction results, a 3D point is only considered if away from a $5\,\text{cm}$ radius from any other previously selected point. Each 3D point is scored for interaction according to an objective function and then sorted in the motion plan according to it. The objective function is composed of three weighted parcels, whose weights sum up to one and have been defined empirically. The first parcel builds from the intuition that bending higher areas of the object (e.g., leafs) is usually easier than their lower areas (e.g., log). Therefore, the first parcel is defined as the inverse of the height of the 3D position in question. The second parcel builds on the intuition that the object's centroid should be the most dense and, thus, difficult to bend. As a result, the second parcel is defined by the inverse of distance from the point in question to the object's centroid. Finally, the third parcel is defined as the density of points in the neighbourhood of the position in question. The higher the density, the more likely the position is of belonging to the most difficult-to-bend object's part.

Fig. 4 shows that the set of most priority interaction points in a typical situation, i.e., with score above 0.7, do correspond to the portions of the object that are more likely to block the antenna. These points are located in the central and denser region of the object. A random point selection policy could lead the robot to perform uninformative interactions first, such as pushing the leafs of a bush. These are non-obstructive despite the fact that the bush itself might be.

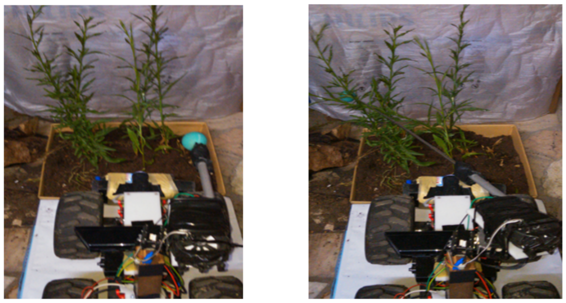


Fig. 5. Typical haptic interaction execution. The antenna stretches to the distance of the furthest interaction point then follows the plan. Note that this behaviours results in a scanning pattern that bends traversable obstacles.

F. Haptic Interaction Plan Execution

If the confidence level on the memory recalled classification is high, then the physical validation by haptic interaction can be shortened. Conversely, very low confident classifications demand for a thorough physical validation. This principle is implemented by exploiting the fact that the sequence of points to analysed, i.e., motion plan, is ordered by relevance. Concretely, only the m-first points are considered, given that m is the position of the point whose interaction score is below $\alpha \cdot c$ and α is an empirically defined scalar and c is the memory recall's confidence level. With a high α, the system reduces the number of interaction points and favours past experiences, whereas a low α increases the number of interactions, making the system to behave more cautiously. As a result, α can be used by a higher-level reasoning system to adapt the speed-accuracy of the system depending, for instance, on environmental stress. Haptic interactions are slower but more accurate than visual cues.

Once the interactions have been selected, the robot proceeds with the actual haptic interaction. The interaction is made by picking the rightmost point found and successively moving the antenna to the closest point, until all it reaches every point. If throughout the process the antenna gets blocked, then the object is labeled as non-traversable and traversable otherwise. To raise the chances of getting the antenna blocked due to the presence of a non-traversable object, the points to be reached are translated towards the object's centroid by an amount of 5 cm. The motion resulting from following the plan scans the environment in a way that raises the chances of getting the antenna blocked when in the presence of a non-traversable object that it would have if the points were sequentially pushed by the antenna. Fig. 5 depicts such a typical haptic interaction.

G. Environment Change Detection

If the object in front of the robot is considered traversable, either via highly confident memory recall or via haptic interaction, the robot will try to traverse it. If the object is in fact a large extension of vegetation, then at each new step of its progression, the robot will recurrently face the same object. Thus, it is thus necessary to provide the robot with a mechanism that hampers it from reassessing the same object at each new progression step. Gating the assessment process until the robot's surroundings change considerably does this. Due to the robot's small size, surrounding vegetation covers most of its sensor's field of view, thus, the changes to be captured are scene-wise. As a result, a global descriptor of the scene, known as *gist*, is adequate.

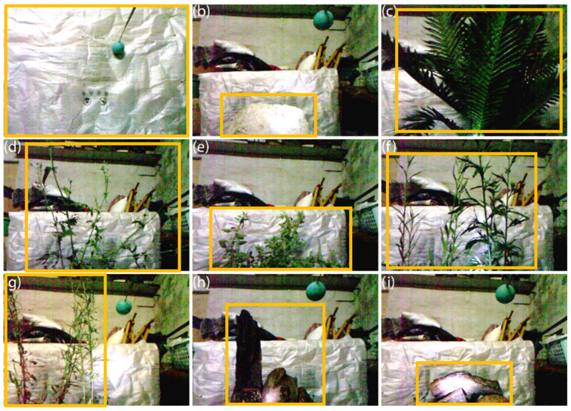


Fig. 6. Objects used for classification accuracy analysis in a controlled environment, as seen from the robot with its depth sensor. The yellow rectangles represent the objects' bounding boxes. (a) Wall (non-traversable); (b) Rock (non-traversable); (c) Big plant (non-traversable); (d) Shrub (traversable); (e) Small shrub (traversable); (f) Tall plants (traversable); (g) Tall Plant (traversable); (h) Vertical logs (non-traversable); (i) Horizontal logs (non-traversable).

A scene's *reference gist* is computed before the robot starts traversing the object. Then, at each progression step, the reference gist is compared against the current scene's gist. If they differ greatly, then the robot is taken as out of the object and the gating process can be unlocked. The gist descriptor is simply the point cloud's overall density, which is computed as the ratio of points obtained from a down-sampled version of the point cloud over the number of points in the original point cloud. Two gist descriptors are said to differ if their associated point densities differ more than an empirically defined threshold, ρ.

III. EXPERIMENTAL RESULTS

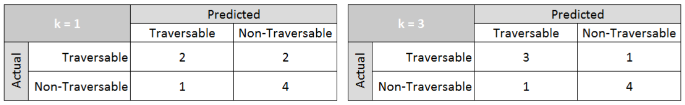
A. Classification Accuracy from Haptic Interactions

To assess the system's classification accuracy based on haptic interactions, the robot was asked to move forward in a controlled environment until an object is found. Throughout the process the robot faced 9 different objects, (a)...(i), one at a time. Let us call these 9 objects *data set 1*. With each of them, the robot engaged on the interaction process so as to determine whether the object is traversable or not. The set of objects includes four traversable plants, a non-traversable wall, a non-traversable plant, two piles of non-traversable logs, and a non-traversable rock (see Fig. 6).

Table I shows the number of interaction points within a given score interval (see Section II-E) selected by the system for each tested object. The table shows that the number of points grows from higher scores to lower scores. This is consistent with the intuition that the good interaction points are fewer than the poor interaction points. Fig. 4 illustrates this phenomenon on a typical vegetated object. As expected, denser objects, such as logs and rocks, tend to exhibit a higher number of high scoring points than thin vegetation. For all objects, interacting with points with a score above 0.7 was enough to give a correct traversable/non-traversable classification.

Score	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
$[0.9, 1.0]$	3	0	0	0	0	0	0	0	0
$[0.7, 0.9[$	5	3	2	1	3	2	2	4	2
$[0.5, 0.7[$	11	2	14	11	4	14	10	2	2

TABLE I. NUMBER OF POINTS SELECTED FOR HAPTIC INTERACTION WITHIN A GIVEN SCORE INTERVAL.



k = 1		Predicted	
		Traversable	Non-Traversable
Actual	Traversable	2	2
	Non-Traversable	1	4

k = 3		Predicted	
		Traversable	Non-Traversable
Actual	Traversable	3	1
	Non-Traversable	1	4

Fig. 7. Confusion matrix obtained from leave-one-out cross-validation.

B. *Classification Accuracy from Learning*

To assess the robustness of the object descriptor (see Section II-C) and the memory recalling process (see Section II-D), a leave-one-out cross-validation analysis was undertaken based on the 9 objects. The principle used is to leave one of the objects out of the training set and then classify it based on the training set, which has been hand-labelled. As depicted in Fig. 7, the system produced a correct traversable/non-traversable classification $67\,\%$ of the times for $k = 1$ and $78\,\%$ of the times for $k = 3$. This an interesting result given the lack of redundancy present in the data set. That is, for $k = 3$, the system recognises the objects based on their intra- and inter-class resemblance.

To evaluate the ability of the system to incorporate new knowledge on the top of the 9 already known objects, the robot was asked to travel towards two unseen objects (see Fig. 8). In a first test, the robot approaches the first object from various angles and for each approach it tries to recall it from memory. The memory grows with the result of each of the new interactions. Fig. 9 shows that the system managed to recognise the object with a tendentially growing confidence as the number of interactions unfolded. The variability in the confidence level results from the fact that in each approach the object looked different to the robot - the object is anisotropic and the depth sensor is impinged with considerable noise. After 20 encounters with the first object, the robot was presented for the first time to the second object, which resulted in a low confident classification. As for the first object, the system managed to recognise the second object with a confidence level that tendentially grew with the number of encounters. Also as for the first object, the second object was approached from various angles. Let us call the several samples obtained from the novel two objects *data set 2*.

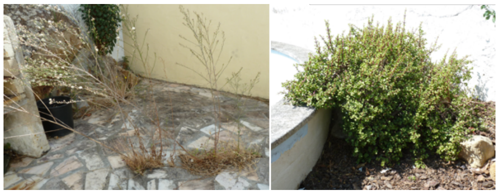


Fig. 8. Novel traversable (left) and non-traversable (right) learned objects.

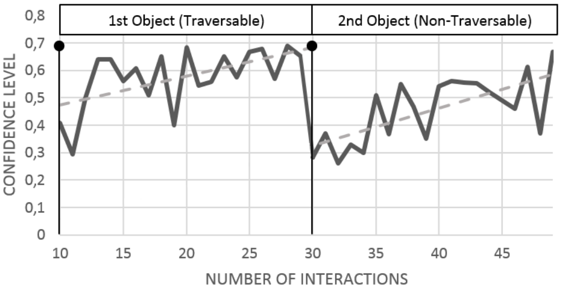


Fig. 9. Classification confidence when progressively incorporating two new objects into memory. The two interrupted lines represent the linear regression for the confidence level before and after meeting the second object.

Object	Traversable	Confidence	Classification
(a)	No	0.42	Not Traversable
(b)	No	0.37	Traversable
(c)	No	0.33	Not Traversable
(d)	Yes	0.65	Traversable
(e)	Yes	0.48	Traversable
(f)	Yes	0.49	Traversable
(g)	Yes	0.60	Traversable
(h)	No	0.44	Traversable
(i)	No	0.23	Traversable

TABLE II. CLASSIFICATION OF OBJECTS IN DATA SET 1 GIVEN KNOWLEDGE ABOUT OBJECTS IN DATA SET 2 WITH $k = 5$. MIS-CLASSIFIED OBJECTS: (B), (H), AND (I).

Let us now assume that the robot's memory is filled with the samples from the two novel objects, i.e., data set 2. Let us also assume that the robot is unaware of the original 9 objects, i.e., data set 1. In this case the robot is said to be knowledgeable of an environment composed of objects contained in data set 2. In a real situation, when entering a new environment, the robot will progressively find new objects that must be capable of classifying and, potentially, integrate in its knowledge base. Table II shows that in most situations the robot is capable of properly classifying novel objects from data set 1 as traversable or non-traversable, given its prior knowledge about different objects from data set 2. This owes greatly to the redundancy in the appearance of objects in natural environments. Interestingly, erroneous classifications are also low confident, which forces the robot to interact with its haptic actuator to carefully assess the actual navigation affordances of the object.

C. *Haptic Interaction Plan Execution*

Low classification confidence compels the robot to engage on haptic interactions so as to robustly classify the object as traversable or non-traversable. The lower the confidence the higher the number of haptic interactions are engaged. This relationship is scaled by an empirically determined scalar α (see Section II-F). To provide some intuition about the parameterisation of this scalar, Fig. 10 shows its effect on the number of haptic interactions. The plot was built by varying the number of samples from object 1 of data set 2 provided to the robot. This variation emulates the effect of learning from various interactions. The higher the number of samples in memory the higher the confidence on the classification and, hence, the fewer the required haptic interactions. The figure also shows that the higher the value of α the more the system values memory over haptic interactions. As expected, α shows itself as a good modulator for the speed-accuracy trade-off.

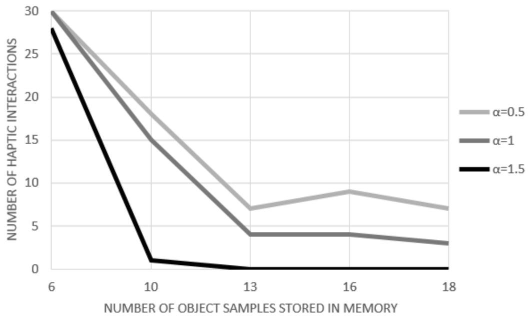


Fig. 10. Impact of different α on the number of haptic interactions.

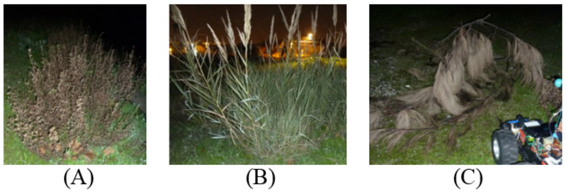


Fig. 11. Objects used for environment change detection tests. (A) Small shrub; (B) Flimsy canes; (C) Twigs with thin leaves.

D. Environment Change Detection

To avoid repeating haptic interactions while traversing a given object, the robot determines an environment density change before reconsidering a new haptic interaction (see Section II-G). Fig. 11 depicts three objects used to assess this capability with a density change detection threshold of $\rho = 0.03$. For this test, the robot was asked to move across the object. To do that, the robot meets the object, performs a haptic classification, which returned traversable for all cases, and then tries to traverse the object. While doing it, the robot evaluates periodically if an environment density change occurred. If it occurs the robot stops and performs a new haptic verification.

Two situations were studied for object A. In the first situation, A-1, the robot met open space when leaving the object, whereas in the second situation, A-2, the robot met an introduced wall-like object. In both situations the robot detected the density change, i.e., from object to open safe and from object to wall. When traversing object B the robot got stuck hampering it from progressing across the object. Correctly, the system remained without reporting any environment density change. As the robot gets stuck it becomes clear that the object is non-traversable despite did not look like it in the first haptic interaction. Corrective measures should be triggered correspondingly. Object C offered no difficulties to the robot resulting in a fast traversal and easy change detection. These results are summarised in Table III and they show that environment density is a simple yet effective metric for change detection in the context of object traversal.

Object	1st Eval.	2nd Eval.	3rd Eval.	4th Eval.	Change Detected
A-1	0.019	0.013	0.014	**0.055**	Yes
A-2	0.011	0.009	0.021	**0.031**	Yes
B	0.020	0.018	0.008	-	No
C	0.014	**0.059**	-	-	Yes

TABLE III. ENVIRONMENT DENSITY CHANGE DETECTION RESULTS.

ACKNOWLEDGEMENT

This work was co-funded by ROBOSAMPLER project (LISBOA-01-0202-FEDER-024961) and by the CTS multi-annual funding, through the PIDDAC Program funds.

IV. CONCLUSION

A ground vehicle capable of exploiting haptic cues to learn navigation affordances from depth cues was presented and validated on a set of field trials. For acquisition of haptic cues the robot employs a low-cost pan-tilt telescopic antenna, whereas for distal sensory feedback the robot recurs to a low-cost depth sensor. Although not limited to them, the simplicity of the proposed system allows its application in small sized robots, which are useful tools for domains like environmental monitoring and search & rescue. These domain applications require from robots the ability to cope with the unstructured configuration of natural environments. This challenge is mitigated by the incremental learning of perceptual skills ensured by the proposed system. We are currently analysing alternative depth descriptors, machine learning mechanisms, and haptic motion planning and execution policies. We are also addressing the set of challenges related to the application of the proposed system to robotic platforms endowed with different manipulation and sensory modalities.

REFERENCES

[1] S. Lacey, J. Hall, and K. Sathian, “Are surface properties integrated into visuohaptic object representations?” *European Journal of Neuroscience*, vol. 31, no. 10, pp. 1882–1888, 2010.

[2] J. Gibson, “The concept of affordances,” *Perceiving, acting, and knowing*, pp. 67–82, 1977.

[3] R. Detry, E. Baseski, M. Popovic, Y. Touati, N. Kruger, O. Kroemer, J. Peters, and J. Piater, “Learning object-specific grasp affordance densities,” in *IEEE Intl. Conf. on Development and Learning*, 2009.

[4] D. Kim and R. Möller, “Biomimetic whiskers for shape recognition,” *Robotics and Autonomous Systems*, vol. 55, no. 3, pp. 229–243, 2007.

[5] J. A. Wijaya and R. A. Russell, “Object exploration using whisker sensors,” in *Australasian Conf. on Robotics and Automation*, 2002.

[6] E. Uğur and E. Şahin, “Traversability: A case study for learning and perceiving affordances in robots,” *Adaptive Behavior*, vol. 18, no. 3-4, pp. 258–284, 2010.

[7] M. Bajracharya, A. Howard, L. H. Matthies, B. Tang, and M. Turmon, “Autonomous off-road navigation with end-to-end learning for the lagr program,” *Journal of Field Robotics*, vol. 26, no. 1, pp. 3–25, 2009.

[8] C. Wellington, A. Courville, and A. T. Stentz, “A generative model of terrain for autonomous navigation in vegetation,” *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1287–1304, 2006.

[9] G. Azzari, M. L. Goulden, and R. B. Rusu, “Rapid characterization of vegetation structure with a microsoft kinect sensor,” *Sensors*, vol. 13, no. 2, pp. 2384–2398, 2013.

[10] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *Proc. of the IEEE ICRA Workshop on Open Source Software*, vol. 3, no. 3.2, 2009.

[11] R. Rusu and S. Cousins, “3d is here: Point cloud library (pcl),” in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011.

[12] R. M. Haralick, H. Joo, D. Lee, S. Zhuang, V. G. Vaidya, and M. B. Kim, “Pose estimation from corresponding point data,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 6, pp. 1426–1446, 1989.

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