Author's Response

December 15, 2017

We thank the reviewers and editor for their insightful comments: we had been working on some of the raised issues even before receiving the reviews, and we could integrate the results of the work and additional experiments in the revised version. As a result, we deeply reworked the paper and we think this addresses most the raised points.

First, we respond to the editor comments in depth. Then, we go through each individual review, and point to relevant changes in the document. Additionally, in the supplementary material file we attach a compressed version of the new manuscript highlighting the changes with respect to the first submission.

1 Editor comments

> This paper addresses learning of ground traversability based on simulations. The proposed method exploits convolutional neural networks (CNNs) to learn traversable regions based on classifiers using generated height-map in simulation. The proposed method is applied to path-planning to balance between the distance and traversability in a real environment to validate its effectiveness.
>
> The paper deals with a relevant problem of navigation using CNNs that are now very intensively studied. However, the reviewers raise two common main concerns.
>
> First, the contributions of the paper over previous work are not clearly stated as Reviewers 7 and 13 point out. The authors need to clarify the advantages of the proposed method over existing work. Although "contributions" are noted in Section I, they just say what are done: the readers cannot see what are the advances and advantages compared to previous work in Section II. Authors are therefore strongly advised to clarify the contributions.

We rewrote from scratch the related work section, which is now much more complete; we follow a structured approach to discuss related work, first dealing with the issue of input (appearance vs geometry), then discussing how other learning-based methods acquire their training set (for traversability, they all use real-world data).

This allows us to explicitly discuss how our approach differs from existing literature; in particular: 1) for the first time we are using deep nets on heightmaps for traversability estimation (whereas deep learning on appearance data has been used in many previous works); and 2) we are the only approach using learning traversability from simulations on procedurally-generated training data (whereas the manipulation and locomotion literature makes massive use of simulations to generate training sets).

We thoroughly discuss advantages and drawbacks. We also added the related literature suggested by reviewers, that was missing from our original paper.

> The other issue is the physical plausibility of the classifier learned only by the simulation with 3-D geometry and its applicability to real situations. Reviewers 7 and 8 wonder whether the proposed method is reliable without taking into account physical properties such as friction and terrain material parameters. As Reviewer 8 points out, there is little support for the claim "We assumed that the terrains 3D shape is the only factor influencing traversability; Our framework could handle these factors provided they can be simulated: ..."

Applicability to real situations is indeed a key issue and we understand that our sentence "We assume that the terrain's 3D shape is the only factor influencing traversability" was misleading: in fact, it was meant as a clear delimitation of the scope of the work rather than as a general statement. We now deal with the issue in much more detail.

The revised paper keeps the same scope and limitations as the original: we are still using only training data acquired from simulation, and we are still considering only the 3D terrain shape as input, assuming solid ground with no issues of friction. In the second part of Section II, we thoroughly discuss the issue, and we elaborate on that more in the Discussion Section, in view of experimental results. Below, we elaborate on the issue.

Factors influencing traversability.

- The 3D shape of a terrain is one of the factors influencing traversability. Other factors such as friction properties, terrain deformation, moving rocks, sand, etc. also affect traversability.
- In this work we focus on the terrain's 3D shape and ignore all other factors. We argue that this makes sense in some common real-world scenarios, such as: grassy slopes, trails in dry conditions, or man-made environments (wheeled robots on roads, wheelchairs). In these scenarios 3D shape is the most relevant aspect influencing traversability. Figure 1 illustrates some of these situations.
- We acknowledge that many other real-world scenarios, such as snowy/icy conditions, sand or mud, are dominated by the remaining factors,

Figure 1: Some situations where terrain geometry is the most relevant feature influencing traversability, often in a non-trivial way.

which our approach does not yet handle. We had not been explicit enough about this in the original submission: we now elaborate this limitation in Section II and V in the discussion sections.

Future options for handling factors other than 3D shape

- In perspective, handling factors other than 3D shape within our approach is possible as long as we can provide training data to the system: this requires to either run simulations in which these factors are accurately simulated, or to acquire training data from the real world. The former approach is becoming feasible as new efficient GPU-based robot physics simulators are released (e.g. Nvidia's ISAAC platform, which should gain wheeled robot support in the next few months); the latter approach requires to acquire training data from the real world. Furthermore, one could argue that 3D data alone is not enough: a gentle grassy slope is traversable uphill, but a gentle snowy slope is not, and they have the same 3D shape. Extending our approach to handle terrain appearance (in addition to 3D shape) only requires to add one input map to the neural net, or to foresee different models for different surface types (which could be previously classified as shown in [1]).

Learning from simulated training data and validating on real-world data.

- In our paper we only use training data obtained from from simulations run on synthetic, procedurally-generated maps; this is both a limitation and a deliberate choice. In fact, one of our goals is to investigate whether the knowledge learned from synthetic maps can be applied to real-world maps.

- We answer the question by simulating the robot on six maps acquired from real-world environments, resulting in plenty of quantitative metrics.
- For two of these six maps, we have access to the real environment: there, we validate our approach by running the real robot in such two real environments (one indoor, one outdoor), which yield qualitative insights.

Future options for using real-world training data

- Using real-world experience rather than (or in addition to) simulated experience as training data fits our conceptual framework without any change. Acquiring real-world training data (i.e. a collection of traversable and non-traversable patches) is possible as long as the system knows the 3D shape of each patch that is being traversed and has accurate ego-motion information (e.g. from a visual odometry pipeline), which allows the system to label each patch as either successfully traversed or not. We recognize that implementing such a system is a significant engineering effort, which we are starting to tackle but that will require many months of development.
- Once one has both (large amounts of cheap) simulated and (limited amounts of expensive) real-world data to learn from, an interesting and relevant research question is: how to optimally harvest knowledge from both datasets? This is a long-term research interest of ours, and is not a subject of the current paper, which however acts as a foundation to such future research line. In this framework, one can see that simulated training data plays an important role. Training a traversability classifier from scratch with only real-world data may be unfeasible: in this context, simulations allow one to extract general bootstrap knowledge (such as: steep slopes are harder than gentle slopes; high steps are harder than low steps, excessive inclinations cause capsizing) that is then refined by real-world data.

A clear discussion of the issues mentioned above is now provided in Sections II and V.

> Reviewer 8 and 13 remark that the experiments are indoor and too simple and trivial. It is advised to show experimental results in more complex outdoor environments. The reviewers are aware that the paper tackles a very challenging problem, but the authors must absolutely answer to those questions about physical plausibility and applications to a real scenario, and this may also lead to clarifications of the originality of the work.

We significantly extended our experimental validation on real-world data. Compared to the original version of the paper, we added three scenarios, one of which is a complex outdoor scenario (Slope: an uneven, bumpy grassy slope flanked by an uphill cement path) on which we could verify our path planning results with the real robot.

The revised paper reports the following experimental results for real-world environments (we also provide this summary in Table I):

1. We consider 3 very different outdoor (Quarry, Sullens (new), Gravel-pit) scenarios plus one indoor (ETH-ASL (new)) scenario. For all these scenarios, we quantitatively evaluate the performance of traversability estimation using as ground truth information obtained in simulation (e.g. whether each patch is actually traversable when the robot is simulated on the map). This positively answers the research question: “can we learn traversability of real-world maps using training data from synthetic maps, under the assumption that only 3D shape plays a role?”. However, since we do not have access to these environments, we are not able to confirm whether these paths are indeed traversable by the real robot. However, we observe that the paths with maximal-traversability computed by our system are coherent with the traversable areas (Fig. 2): they are the paths that an operator would follow if we were teleoperating the robot between the two points (e.g, they stay on roads and avoid uneven terrain whenever possible).

2. We kept (but summarized for space reasons) the simple indoor experiment that was in the original paper (Bars map): the experiment confirms that the learned criteria for the traversability of a bar-like obstacle on the ground match the observed capabilities of the real robot. The full experiment description is left in supplementary material.

3. Most importantly, we added an experiment (Slope map) running the complete pipeline (including our own RGB-D mapping using a hand-held Tango sensor) on a complex outdoor environment covering about $40m^2$ (Fig. 3. The results match the results we had obtained in the quarry map; in this case, since we have access to the environment, we could verify that the maximal-traversability path between two distant points (square mark) is actually traversable by the robot; the maximal-traversability path to a difficult-to-reach point (star mark), which was estimated as traversable with a 21% chance, is indeed not traversable (due to a step that is slightly too high for the robot to pass) even though it qualitatively looks a rational choice; the maximal-traversability paths to points on the grassy slope (circle mark) tend to follow the smooth paved path uphill, and then descend on the grass, rather than attempting to directly ascend the grass, which would not be traversable.

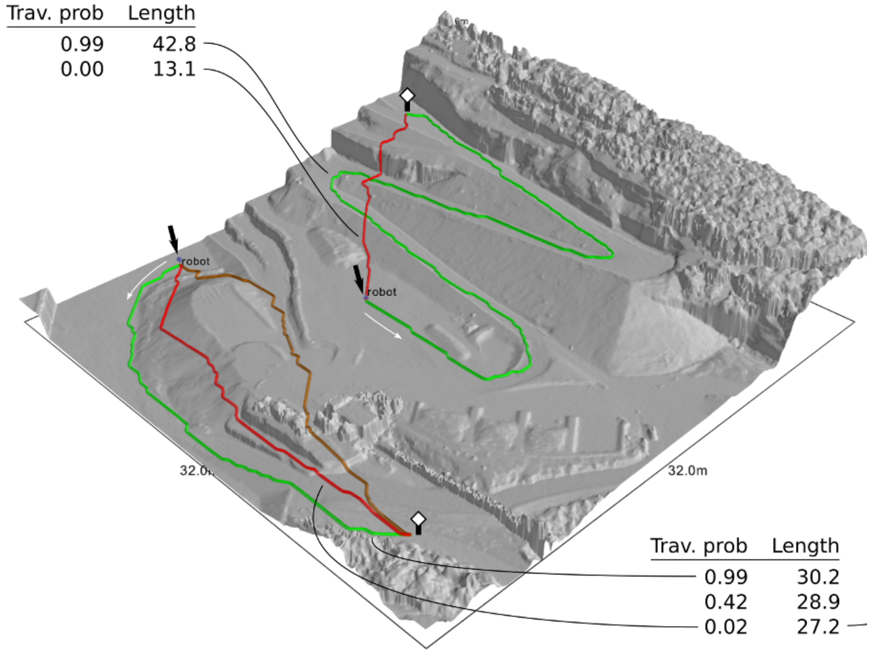

Figure 2: Examples of paths found with our traversability estimation approach in the Quarry map.

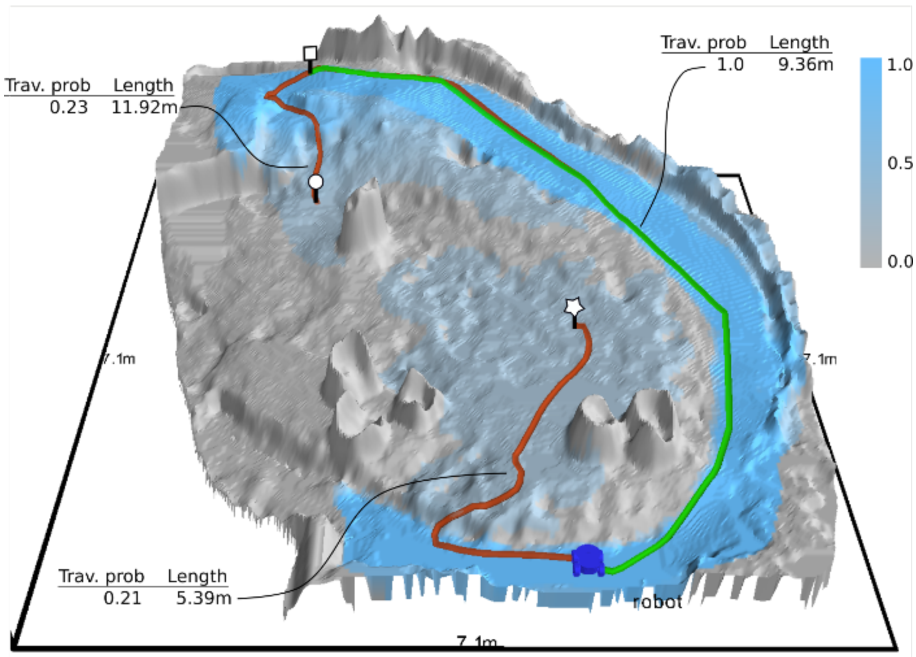

Figure 3: Paths found in the Slope map and verified in the real world.

> Other important technical issues are also questioned: property of noise in simulation (Reviewer 7)
>
> Reviewer 7: 1) The authors are using Perlin noise to simulate the terrain geometry. However, Perlin noise does not have diversity in the large scale, i.e., the patches of Perlin terrain would have similar statistical property in 3D geometry. Thus the training dataset would not be able to represent all the different 3D terrain geometries. Thus the proposed method seems to be difficult to handle terrains with diversified geometry.

We clarified the text, and provide an in-depth answer to reviewer 7 below in this letter.

> possible discontinuity of steerable orientations among patches (Reviewer 8 and 12)

The implementation in the original paper assumed that the robot could always rotate in place, which was an excessive simplification as this is not always true in practice. The revised version of the paper now considers a classifier that given a patch, estimates the probability that the robot can successfully rotate in place 45 degrees clockwise; another classifier is trained for predicting whether turning 45 degrees counter clockwise is possible; in Section IV.D we describe the application of such classifier and report quantitative values for its performance.

When planning paths, all steering happens via in-place 45-degrees rotations, and the probability of successful rotation is accounted for in the path traversability.

Note that the paths computed in the quarry dataset did not change as a result of this, because they were already passing through traversable ground that allows in-place rotation (plains, slopes, short bumps).

> and computation time (Reviewer 13).

We now discuss computation time in Section IV.C, and most importantly compare our results with the time that would be needed when densely simulating a robot on a map.

> Those issues should also be clarified, as well as including and discussing lacking relevant work (Reviewer 8).

We integrated most of the suggested relevant literature (thanks!) in the related works section.

> The authors are encouraged to revise the paper to address the insightful comments from the reviewers to reach the quality for publication in IEEE RA-Letter.

The reviewers' and editor's comments allowed us to significantly improve the document quality in this revision. We would like to thank you all for your precious contribution.

In the following, we report reviews of each reviewer and answer pointwise questions.

2 Reviewer 7

> This paper presents a method to determine the traversability of a terrain using two-class classification. In particular, the authors assume that the 3D geometry is the only factor influencing the traversability, and then learn a classifier which determine whether a given patch of terrain is traversable given the robot's move direction. The training data for the classifier is from a simulation: given a terrain, the simulator determines whether a robot can traverse a patch at a given position and along a given direction. The learned traversability is then used to compute an optimal trajectory which balances the length and the traversability.
>
> The main limitations are discussed by the authors: it assume the traversability is not related to terrain's physical factors like friction, and does not take into account the velocity/acceleration of the robot.

We extended the discussion on this topic in the paper (and covered the topic in more detail in the editor response above). The paper now reports additional experiments on 3 new outdoor real-world maps (one of which features a real-robot validation of the computed paths).

> There are some other issues about the simulation:
>
> 1) The authors are using Perlin noise to simulate the terrain geometry. However, Perlin noise does not have diversity in the large scale, i.e., the patches of Perlin terrain would have similar statistical property in 3D geometry. Thus the training dataset would not be able to represent all the different 3D terrain geometries. Thus the proposed method seems to be difficult to handle terrains with diversified geometry.

Indeed, plain Perlin (or simplex) noise by itself would not yield enough variability in the training data, even when mixing multiple spatial frequencies and amplitudes. In our procedure, however, we could generate a large amount of variability by adopting four techniques:

- By exclusively using a plain Perlin noise map (or sums of Perlin noise maps), we would never obtain important structures such as steps, rails, holes, stone-like bumps surrounded by flat ground, etc. Let m be a

Perlin noise map with a period of a few meters, and assume its range is [-1, +1]. We apply a scalar function to the values of m to convert them to height values. If such function is $f(x) = x \cdot k$, we obtain smooth hills with height $2 \cdot k$; if such function is $f(x) = \text{sgn}(x) \cdot k$, we obtain a curved sharp step (tracing the locus of points in which the value is 0) with an height of $2 \cdot k$, connecting a flat upper surface with a flat lower surface; if the function is $f(x) = \max(x, 0) \cdot k$ one obtains bumps lying on a flat ground.

- Each map is given by the sum of two such functions, each applied to a different Perlin noise map.
- Within each map, one parameter (such as, for example, the height of steps) smoothly changes along the x axis, whereas another parameter (such as, the curvature of steps) smoothly changes along the y axis. This ensures that a single map will represent a large amount of variability, and that in at least a subset of the map the robot will find "interesting" terrain that is neither too trivial or too challenging.
- We use 30 different training maps, each generated with different functions and ranges of parameters. This yields a total training area of 3000 m^2 which represents a large variety of different structures.

We improved our discussion of the generation technique in order to make it clearer for the reader. However, we understand that an exhaustive and motivated discussion of the topic would take at least a couple of pages, but we don't think that it's worth to use so much space: procedural generation is a large research field, and our approach is just a fast-forward way to obtain interesting training data. Our aim in the paper text (Fig. 4) remains to provide an intuition to the reader. We provide in addition, as downloadable supplementary material:

- High resolution, annotated pictures of some of the maps (akin to those reported below)
- Raw data for all generated heightmaps and code for generating them
- An appendix explaining the adopted functions and parameters

> 2) The simulated noise and the actual noise of the quarry dataset seems to be difficult. Why the model learned from the synthesized data could generalize well to the quarry data?

As outlined above, the synthetic datasets represent a large variability of terrain structures. Moreover, training datasets are very large (450k samples), and the CNN architecture we use is relatively simple, so it is expected

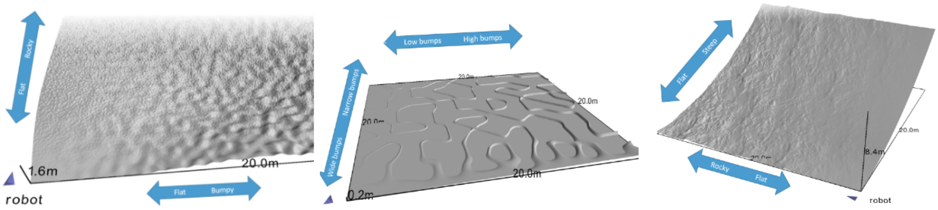

Figure 4: Examples of procedurally generated heightmaps and some of the parameters used to represent challenging terrain patterns.

that this yields good generalization capability on unseen terrains. However, we agree that this outcome was not a foregone conclusion and that this represents an interesting result of the paper.

3 Reviewer 8

> This paper provides a framework for a vehicle that traverses an environment composed of traversable and non-traversable regions. The authors propose training the classifier using extensive geometric simulations and compare a feature-based classifier with a CNN classifier.
>
> The paper is clear, well written and well organized. The problem is very challenging and the inclusion of a CNN classifier in the process of path planning is very sensible, particularly if it allows the inclusion of real experiments as training data. I believe the paper is very suitable for a RA-L publication, but the experimental results are still indoors and do not address the outdoor challenges.

Including real-world experiments as training data is (conceptually) straightforward in our framework; we discuss the issue in Sections II (second part) and V.

The revised version of the paper extends the real-world validation, including a new outdoor experiment with a real robot, even though we still use simulation-only training data.

We discuss both points in detail in the *editor comments* above.

> In the related work, the authors should make a statement (or many) in which point(s) your work is different from the related work. Try to be very clear where do you advance the state-of-the-art in relation to the works that you cited in Section II.

We clarified our original contributions, and reworked the related work section to make sure to point out how our approach extends previous work. We also added several related papers suggested by reviewers, that were missing from our original paper. We discuss the issue in detail in the *editor comments* above.

> In my opinion, the main weakness of the work is that the traversability is given by a simulator. In real problems in field robotics, it is well known that accuracies of such simulators are very poor. There is a large literature that investigates the use of analytical models [e.g. Estimation of track-soil interactions for autonomous tracked vehicles, Anh Tuan Le, David C Rye, Hugh F Durrant-Whyte, ICRA 1997][body of work from Sanjiv Singh on autonomous excavation, etc.]. The real experiments are still in the indoors setting, with well-defined floor and a simple geometric obstacle. A real outdoors scenario—which, in the end, seems to be the main motivation of the paper—offers additional problems that are not addressed by the simulator, such as the slippage, soil granularity, uncertain friction, soft (traversable) obstacles, loose gravel/rocks, etc.

Yes, our approach has limitations and we made sure to be explicit about those in the revised version of the paper. We deal with the issue in depth in the "editor comments" section above.

The revised version of the paper includes an outdoor real-robot experiment on a grassy slope that, even though limited in scope, provides some evidence about the utility of the approach, at least in scenarios where slippage or soft ground would not be an issue.

> Since the authors are using a learning approach based on CNN, it would be interesting to investigate how to incorporate the use of real data instead of simulated data. For example, by applying discrete commands on the real robot in order to collect training data.

Good point: we discuss this point briefly in the revised paper and with more detail in the *editor comments* section above.

> The path planning in IV.D could be explained in more detail. It is intriguing that the authors simulate the robot without steering, but in the real experiments, the robot does steer. Thus, I assume the path planner is doing the steering. Do you treat the problem as a holonomic path planning?

In the original paper we assumed the robot could always rotate in place. In the revised version we train a classifier that predicts whether the robot is able to turn in place (clockwise or counter-clockwise) on a specific patch. See the discussion in the *editor comments* section.

> I am also curious what would happen if the transitions between two patches require opposite orientations to satisfy traversability. More generally, how does the framework handle the transitions between action if they generate discontinuities (that is, if the orientations, velocities, motor commands, etc. are not the same)?

For orientations, the robot rotates in place at 45 degrees steps. Discontinuities in velocities or motor commands are disregarded, as the robot is assumed to move slowly (which is realistic for wheeled/tracked off-road robots or legged robots), which allows us to ignore the dynamic aspects of the robot's motion (otherwise, the robot's instantaneous speed should be an additional input other than the 3D shape of the patch: this is now mentioned in Section V).

> Please, check this work [Non-parametric learning to aid path planning over slopes, Sisir Karumanchi, Thomas Allen, Tim Bailey, Steve Scheding, IJRR 2010] as it may be well aligned to your interests (although I don't remember the details of this particular paper).

Thanks, in the new related works section (rewritten from scratch), we included the suggested literature, which is indeed relevant.

> I tend to disagree with the way the paper treats more challenging/realistic problems as mere extensions of the method. For example, the sentence "We assumed that the terrain's 3D shape is the only factor influencing traversability; other factors may also be considered, such as compactness, friction, and instability. Our framework could handle these factors provided they can be simulated: for example, if terrains whose 3D shape suggests a loose gravel surface were simulated with lower friction," indicates that a huge challenge is still to come because simulations of such realistic scenarios will be certainly very difficult and a source of mismatch. The same applies to moving from a simple skid-steer robot to the problem of locomotion.

Yes, our discussion of the potential extensions was overly shallow / simplistic in the original paper. We present the issue in a more realistic way in the revised version (by necessity, synthetically): see the *editor comments* section for a detailed discussion of the changes.

4 Reviewer 12

> The paper presents a framework for learning ground traversability for mobile robots. It relies on extensive data collection from simulations and utilizing deep learning to classify ground patches as traversable or not. Main components of the framework are (1) procedural generator for heightmaps which represent simulated terrain for training purposes, (2) CNN based heightmap-patch classifier and (3) orientation based-traversability representation. The results are validated on a real heightmap and with a real robot (Pioneer 3-AT).
>
> The paper is well written, structured and the authors clearly state its contribution. The literature review and related work overview is extensive. It is easy for a reader to follow and comprehend authors' work and results. The multimedia supplement is located on github with a reference (link) in the paper. The

> videos nicely demonstrate the data collection phase, validation on a test data and the real robot experiment.
>
> I only have a few minor comments and questions: 1) Is there only one procedurally generated heightmap (10x10m) per terrain type over which you always randomly respawned the robot or you generated another of the same type after a certain number of respawns?

No, we have 30 procedurally generated training heightmaps (total: 3000 m^2), and the robot respawns to a random map each time, in a random location at a random orientation.

> 2) The current patch is labelled as traversable only if the distance between a future and current pose is greater than a threshold d. That means if the robot tumbles down a slope and satisfies that threshold, a patch will be traversable?

Yes it does, as long as the robot falls in the forward direction and not sideways. And in fact, steep downhill slopes in the quarry map are found as traversable when the robot points straight down (which, by our definition, they are). We explicitly comment about this in the revised paper, and explicitly note that we are not aiming to estimate controllability.

> Because you know robot's speed, you can anticipate which distance it will cover in a certain time window, so why use only a displacement threshold when you could measure the distance between the actual and ideal pose (which the robot would reach on a flat ground) and put a threshold on that value?

Exactly: the threshold is in fact calculated (off-line, once for all) as you suggest.

> 3) Since other researchers might be interested in using your framework for their robots, it would be interesting to mention the computational time required (for simulation part and CNN training).

In the revised version of the paper we discuss computation time in section IV.C, and most importantly compare our results with the time that would be needed when densely simulating a robot on a given map.

5 Reviewer 13

> This manuscript deals with traversability estimation as a height map classification problem. The authors built a convolutional neural network that predicts whether the robot will be able to traverse such patch from left to right. The proposed approach was tested simulation using real-elevation datasets and indoor experiments. The reviewer's commons are follows.

> The reviewer could not find originality/novelty of this paper from Section I and II. Please make clear the difference and advantage of the proposed methodology of this paper.
>
> A traversability estimation framework described in this paper is based on a lot of previous works. The original points are not clear. The description seems just combination of other works.

We clarified our original contributions, and reworked the related work section to make sure to point out how our approach extends previous work. We also added several related papers suggested by reviewers, that were missing from our original paper.

> In the experimental results, evaluation of proposed method seems not enough. It is better to add discussion about accuracy or limitation of proposed method.

We significantly extended the experimental validation section; please see the "editor comments" section above for an in-depth discussion of the changes.

> The experimental setting using an actual robot is quite simple compared with the environment in Fig. 7. Please add explanation about the reason.

The revised version of the paper includes an outdoor real-robot experiment on a grassy slope that has the same characteristics as the quarry map: this experiment provides some evidence about the utility of the approach in scenarios where slippage or soft ground would not be an issue.

> Caption of table I should be upside of the table. There are some typos. Please check the paper carefully.

Thanks for the comments: we fixed the table caption and fixed many typos in the paper.

References

[1] J. Delmerico, A. Giusti, E. Mueggler, L. M. Gambardella, and D. Scaramuzza, "On-the-spot training for terrain classification in autonomous air-ground collaborative teams," in *Proc. of the Int. Symp. on Experimental Robotics (ISER)*, 2016.