Traversability Analysis Using Terrain Mapping and Online-trained Terrain Type Classifier

Henry Roncancio¹, Marcelo Becker¹, Alberto Broggi² and Stefano Cattani²

Abstract—Path estimation is a big challenge for autonomous vehicle navigation, especially in unknown, dynamic environments, when road characteristics change often. 3D terrain information (e.g. stereo cameras) can provide useful hints about the traversability cost of certain regions. However, when the terrain tends to be flat and uniform, it is difficult to identify a better path using 3D map solely. In this scenario the use of a priori knowledge on the expected road's visual characteristics can support detection, but it has the drawback of being not robust to environmental changes. This paper presents a path detection method that mixes together 3D mapping and visual classification, trying to learn, in real time, the actual road characteristics. An on-line learning of visual characteristics is implemented to feedback a terrain classifier, so that the road characteristics are updated as the vehicle moves. The feedback data are taken from a 3D traversability cost map, which provides some hints on traversable and non-traversable regions. After several re-training cycles the algorithm converges on a better separation of the path and non-path regions. The fusion of both 3D traversability cost and visual characteristics of the terrain yields a better estimation when compared with either of these methods solely.

I. INTRODUCTION

One of the key features of any unmanned ground vehicle is the ability to detect navigable areas, such as streets in urban environments or traversable paths in off-road environments. In the literature this problem has been faced with a variety of methods, from 3D terrain analysis [1] based on stereo data to the use of some a priori knowledge of the roads visual characteristics [2]. However, none of them deals with mutable path visual characteristics that actually represent the most common scenario for a AGV moving on a long haul; the data input may be noisy and eventually new for classifiers, leading to possible misdetections.

This paper presents a vision-based path/road detection algorithm for unstructured (e.g. no lane markings) urban and off-road environments, able to adapt its behaviour to a changing environment. This is achieved by *on-line learning* of the environment visual characteristics, based on the path followed so far: we assume that a given region is drivable, with high probability, when the traversable cost of traveling through it is low. On the basis of this assumption we update, little by little, our knowledge on road characteristics, allowing the algorithm to adapt itself and learn from new training examples that are continually provided; as a result, it learns the terrain visual characteristics of traversable and non-traversable

regions starting from the corresponding 3D traversability cost map.

To learn those characteristics we created a terrain type classifier, initially trained with examples provided by a human being (those examples are based on his experience driving in several environments). Then, on-line learning is activated to retrain the classifier with new visual characteristics, getting feedback of the path. Eventually, it will produce a final traversable map based not only on the free space approach (3D terrain map) and the terrain classification approach (a priori knowledge), but also based on the on-line learning classifier. This approach adopts a relatively small initial database of terrain types, letting the on-line process to expand that database by adding the most relevant information about current terrain characteristics.

In the literature, unsupervised learning feedbacks have been proved [3] to be effective implementation of on-line learning for road extraction. Also Zhou et al. [4] implemented an on-line learning of the road: to determine the new training examples their algorithm looked for the largest connected road region; to establish that region, all the pixels were firstly classified in path and non-path (which turns out in a binary image); the resulting image was then processed with morphological operations to join pixels to the same region. Then, they used the simply connected road model [4] to determine the largest one. The main assumption in that approach is that all the pixels in the largest connected road area are part of the road. Despite needing an actual ground-true label to feed back the classifier, that estimation yielded a good approximation to the actual road and non-road regions. The road estimation allowed the algorithm to identify which pixels were misclassified, and therefore, use them as examples to feed back the classifier. Another related paper of Zhou et al. [5] adopts a road probabilistic distribution model, which assigns a weight to each pixel in order to give it an importance for the next retraining cycle; pixels located in the center of the estimated road will receive higher values than those on the borders.

Unlike the largest connected path region used by Zhou et al. as ground-true label to feed back their classifier, our approach relies on the information coming from a 3D traversability cost map (that has been proved to be feasible for classifiers feedback [6]) to connect the pixels and estimate the path. One advantage of starting from 3D traversability map is that it is possible to assume some regions as ground-true labels, i.e. regions with a large cost are likely not road (obstacles for example); conversely, the actual road surface will belong to, or included in, the low traversability cost regions, with high

¹Henry Roncancio and Marcelo Becker are with LabRoM - Mobile Robotics Lab. USP-EESC-SEM. São Carlos, SP, Brazil. roncanciovl@sc.usp.br, becker@sc.usp.br

²Alberto Broggi and Stefano Cattani are with VisLab - Università degli Studi di Parma, Italy. broggi@vislab.it, cattani@vislab.it

probability. To refine the estimation, the initial hint (cost) in each cell is weighted on the basis of the corresponding type of terrain. The type is determined by a classifier (an SVM) of the following classes: asphalt, big gravel, small gravel, soil, grass, bushes and stones. By using this refining, it is possible to deal with those scenarios where 3D traversability costs are similar, such as a gravel path surrounded by grass (i.e. everything is flat and free of obstacles).

In summary, in case of flat terrain free of obstacles, it is difficult to extract valuable information just from 3D traversability costs, hence we complement them with a priori knowledge on terrain types. On the other hand, when there are large differences among the traversability costs of different regions, there is also a way of identifying the traversable and non-traversable regions just from geometrical properties, allowing updating the initial a priori knowledge on terrain types and feeding back the classifier.

The paper is organized as follows: Section II explains some of the methods implemented. Section III describes in general terms the steps for achieving the path estimation. Section IV presents the details related to the on-line learning. Section V shows some tests on data coming from the vehicle. Section VI discusses the advantages and drawbacks of this approach and future work. Finally, some conclusions are presented in the last section.

II. METHODS

A. 3D traversability cost map

The traversability cost map is a 2D representation useful to provide information about terrain traversability of cluttered environments, even for scenarios in which there are no obstacles clearly identified, e.g. slopes. The map is generated starting from a grid of cells, of uniform size, arranged to provide a 3D view of the terrain, where each cell (represented by a cuboid) shows the elevation and inclination of the terrain. This map is built from the disparity space of the stereo image; the resulting 3D points in the space are used to generate a surface model of the terrain. Finally, the surface slope, the elevation of the cell and the obstacles in the cell neighborhood are used to define the cost for each cell [7] [2]. The Fig. 1 shows a 3D view of a traversability cost map.

B. Terrain Classification

The terrain classification is cell-based. To extract the part of the image that the cell is pointing, since the cell is a 3D representation, its position is correlated with the input image by Inverse Perspective Mapping. In this paper, a cell corresponds to an image of 10 x 10 pixels. For each cell a set of texture descriptors are extracted, computed on the basis of statistical parameters measured on the cell's pixels; these descriptors compound the cell's *features vector*.

The statistical parameters used to perform classification are based on tonal features such as mean (mean values in RGB channels, mean and standard deviation of the gray-scale image), variance, skewness, energy, entropy and kurtosis of gray levels, and in addition, autocorrelation features extracted

Fig. 1: 3D Traversability Cost Map: the greener the cell, the lower the cost; conversely, the redder the cell, the higher the cost; The blue cells are unknown non-traversable regions, and the yellow ones are obstacles.

from gray level co-occurrence matrices, known as Haralick features [8]. The resulting feature vector has 14 elements.

C. Inverse perspective mapping IPM

The IPM is used to remove the distortion due to the perspective view in which a scene is acquired and on the distance of the objects from the camera. It compensates the changes in the pixel content in order to improve the classification of all the images (every cell) in the map.

D. Tuned Support Vector Machine (SVM)

In the training process, an SVM is tuned to look for the best Kernel-based parameters that achieve the best accuracy in the prediction. The tuning was made for an SVM with a radial-based function (kernel) [9].

The performance of the path estimation is evaluated using metrics for binary classifiers.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Number\ of\ Samples} \quad (1)$$

Positive results are associated to traversable examples, and conversely, negatives are for non-traversable ones.

III. THE PATH ESTIMATION

Generally speaking, the estimation is compound of two data: the traversability cost of the terrain in front of the vehicle and the terrain visual features. Firstly, the terrain map is generated from the disparity stereo image. The image in each cell of the terrain map is remapped to remove the distortion perspective. This makes sure that the information in the images is independent from the distance from the camera to the terrain regions being observed [10]. The corrected image, called "tile," is processed to extract its feature vector in order to classify it. (This feature vector is used for retraining, provided that the tile is useful to feed back the model.)

One important point on this approach is that the cost in each cell of the traversability cost map is weighted according to the terrain type predicted in the cell (Fig. 2). Because the

Fig. 2: Merged traversability cost computation: it is a combination of 3D map and texture recognition of the terrain.

Fig. 3: Map segmentation steps. The real scene is projected over the 3D map as a small image.

traversability is the main criterion to estimate the path, due to its certainty about the terrain, the predicted terrain type is basically used to modify the "resolution" of the cost values. For traversable terrain classes, the cost value is weighted by values fewer than one, which means the resulting cost will be several times lower than the former one. Conversely, for non-traversable classes, the weight is larger than one, which increases the cost of a cell. The weight is different in each terrain type; for example, asphalt has the fewest weight (0.2), because it has the highest certainty to be traversable; conversely, bushes have the highest weight (5). This weighting enables the algorithm to segment more easily scenarios

in which the costs are similar. For example, in a scenario compound either of asphalt or soil surrounded by grass, in which there is no big difference among the costs, the weighting increases the cost in the grass region and reduces the cost in the asphalt/soil region, so a difference between traversable and non-traversable regions can be found.

The merged map has thus more information to segment the regions. The segmentation is carried out by applying watershed to the image that represents the costs of the traversability map; it is actually done over the traversability costs (Fig. 2). The watershed was implemented as a marked-based segmentation algorithm, i.e., the points in which there are certainty about the

traversability and non-traversability were marked as starting points in the watershed. The final segmentation yields an estimation of the path. Noise is reduced using some morphological operations. The next section shows this segmentation in detail.

The following stage is the estimation of the concordance in the segmentation (Fig. 2); this metric evaluates the coherence between the region segmented and the type of terrain predicted on the cells. Its value is determined through the accuracy of the prediction, assuming that the actual class for each cell is the one found with the segmentation. Because they are not ground-true values, this accuracy is better mentioned as a concordance in the segmentation rather than as an actual accuracy. When the on-line learning converges into a separation of the regions, then the concordance is high. According to our tests, when the concordance is higher than 95%, to try to retrain the model classifier has a contrary effect. Due to the inherent noise of our approach, in which the changes in the path generate misclassified cells, to try to achieve a concordance of 100% disables the learning to converge into a solution. So, the retraining is kept while the estimation is not good, i.e., when the classifier is still learning or when new features have to be learned.

After considering a retraining in the model, new training examples are chosen to complete the database (see the next section). It is worthwhile to mention that a former database provided by an expert is kept through all the learning process. In each cycle, new examples are added and old ones are removed. How the database is updated is also explained in an upcoming section. With the new training set, the SVM is tuned in order to update the model classifier.

A. Segmentation of path and non-path regions

The path estimation is carried out with watershed and morphological operations. Firstly, the traversable cost map is transformed into a gray-scale image in which each pixel represents a cell in the map, and each pixel's gray value is the cost of that cell in the map. A morphological closing is applied to the image to reduce small holes and separate more easily the regions, as shown in Fig. 3. We initially evaluated the flood-fill algorithm, but it cannot sometimes segment all the road because noise in the middle of the road makes the flood stops. This flooding generates too much wrong information to feed back the classifier because some regions are marked as non-road when actually are road (they are not reached). To segment most of the possible road regions, the watershed algorithm was used instead of the flood-filling. In this case, even though there is noise on the road, all the regions with low traversability cost are segmented as road and, subsequently, by using morphological operations they are connected.

Another advantage of the watershed is that one can use ground-true labels as the starting point to define the segmented regions; conversely, when the label of a pixel is uncertain, it is specified that the pixel should be segmented. This is specified with a "mask," in which a label (gray value) for the known pixels and zero for the unknown pixels are set. Fig. 3 shows a mask with three labels (white, dark gray and light gray). For

some pixels we are sure they are actually road, such as the pixels with low gray value (low traversability cost), and for pixels with a larger gray value we are sure they are not road, e.g., obstacles and big objects. For the others, the value in the mask is set to zero. After doing the watershed, there are still isolated regions that are not coherent with its neighbors, such as the white cross in the resulting segmentation showed in the Fig. 3; to reduce that noisy result a morphological opening over the image is applied; as a result, the final segmentation is more accurate.

IV. THE ON-LINE LEARNING

The on-line learning is carried out by feeding back the classifier with new examples as the vehicle follows a traversable path. To know which cells should be included as new training examples, it is needed a first path estimation. This initial estimation enables the algorithm to evaluate the current model classifier, that is, finding the cells that had an "unexpected" prediction, i.e., non-traversable cells predicted as path; and the cells with "missing" prediction, i.e., actual path predicted as non-traversable region. Cells are evaluated by its group (traversable and non-traversable region), so that if a cell was predicted by the classifier as path, but it was not estimated as path, then that cell will be considered a new example for one of the classes in the path region. In fact, cells are not evaluated by their own classes: in this case it would be impossible to use unexpected and/or missing cells, due to the uncertainty of its label. To assign a class to the cell, this approach looks for the most frequent class in each estimated region; this is done by counting the cells according to its class. The new training examples are then labelled to the most frequent class in the region, even though they do not actually belong to that class. The "missing" results are added as training examples for the most frequent class in the traversable region. Conversely, the "unexpected" results are added for the non-traversable class. The new examples yield a way of adapting the model to the actual variations of the regions even though they might be labeled to another class. To reduce bias in the classifier, the training set always keeps ground-true examples provided by an expert. This feedback works for most of cells, except for the ones near the borders, because its class has a high uncertainty, so that those cells, and its neighbours, are not considered for retraining. In addition, since the uncertainty of stereo range data increases with the distance to the camera, the farthest cells in the map are deprecated, as well.

The Fig.4 shows several cycles of the learning process: note how non-traversable region is better defined as long as the classifier is retrained. As a result, after several cycles, the path estimation is improved, as is shown in the third column. When the path estimation reaches good result, the training is stopped in order to avoid algorithm divergence.

The results also depend on the training rate; when the rate is too small the algorithm is biased for the current terrain. When the rate is too large, the changes in the path are not updated. In our test, we took a video with frame rate of 12.5 Hz and

Fig. 4: On-line Learning on the Traversability Cost Map: first column show the terrain classification, where source images are projected over the 3D map; obstacles (the yellow cubes) are analyzed in the 3D point cloud coming from the disparity space; second column shows the merged traversability cost (the redder the cell, the less traversable); third column shows the final segmentation.

each 8 frames the classifier was retrained, if any training was needed.

A. Data Base Updating

The main idea behind the on-line learning is to update the database in every cycle and, consequently, complete the classifier model. However, for each cycle it is also needed to keep the size of the database under a certain limit, since the larger the data, the longer the processing time: hence, old training examples are deprecated to make space for new ones, except those examples provided by the expert, which are always used. To avoid losing critical examples, this algorithm takes advantage of the classifier based on support vectors (SVM). The SVM defines its model using a relevant set of all the examples, i.e., the support vectors. Therefore, after training, the new examples that were not support vectors were eliminated. However, to keep a balance in the data, the same amount of examples for both traversable and non-traversable regions is eliminated; and to avoid bias, they are chosen randomly. The space to be freed depends also on the number of the new examples: when the algorithm is converging, the number of new examples is fewer and fewer; therefore, fewer new examples are eliminated to avoid losing the relevant data. When the on-line learning is activated, we had better results by leaving the new examples unbalanced, but after a few cycles, the training examples should be balanced to allow the

Fig. 5: Data Base Updating

algorithm to converge; otherwise, the data generate a bias in the classifier. Fig.5 shows the updating process; firstly, the features vector are calculated in the cells. Secondly, for each cell is predicted a terrain type and its class is stored together with its feature vector for a posterior comparison. After path estimation the features of the selected cells are retrieved. Finally, they update the former database provided by the expert (compound of 4000 examples). In our tests, the database size increased itself until 5000-5500 examples when the learning was working, but generally speaking it depends on the scenario.

V. TESTS AND RESULTS

The algorithm was tested in the video captured with a Vis-lab's Piaggio Porter, equipped with a Point Grey Bumblebee XB3, 3.6mm focal length and 24cm baseline. In Fig.6 are shown processing results examples of 3 frames: each frame is a compound of 4 images, the original one (up left), the 3D map showing the terrain type for each cell (up right), the merged traversability cost (down left) and the final path estimation (down right). Fig.7 shows a hard scenario: here the road (i.e. asphalt) is covered by quite a lot trees leaves that, by accident, cover also the grass on the traffic isle. Moreover, also the 3D traversability map is pretty flat, since the differences, in height and slope, between asphalt and grass are very low.

VI. DISCUSSION AND FUTURE WORK

The main advantage of this approach is the ability to improve the segmentation of traversable and non-traversable regions by learning of data coming from a approximate path estimation. Having as reference point the cost map, the algorithm is able to get relevant features of the traversable regions in order to recognize them again in the next frames.

Fig. 6: Processing examples: note how in frame (a) the final estimation is strongly affected by the initial 3D traversability map, that tends to uniform all the flat areas; in (b) is shown a mid step scenario, while (c) shows how 3D traversability costs and visual classifications are correctly merged together, to obtain an accurate path estimation

Fig. 7: Difficult scenario: both asphalt and grass are covered by leaves.

This approach worked well for scenarios in which the path is not well defined just by 3D traversability cost (too uniform). In that case, the texture features of the terrain helped to add some additional information in the map in order to identify the actual path. There are still scenarios hard to deal with; for example, when both geometric (height and slope) and visual characteristics of the terrain are uniform; strong shadows and sudden changes of light (e.g. entering tunnel or a wood) can be challenging for the algorithm, since road and off-road can have similar visual characteristics: in that case it is better to rely on the 3D map.

For future works will be considered these points: a method to decide whether the path should be either estimated using the 3D map combined with the terrain type classifier or using the 3D map solely. Other descriptors coming from the 3D map, such as the terrain slope and the elevation standard deviation, might be included as descriptor. Also, a method to reduce the noise in the borders of the segmentation will be tested. When the borders are too noisy, the algorithm start being fed with wrong examples because either the path or the non-path region invades the other one. Lane markings, if there are any, could support the detection of those borders.

Finally, a temporal integration and tracking of the road shape, supported by ego-motion, would increase the detection robustness.

VII. CONCLUSIONS

This paper presents an algorithm taking advantage of on-line learning to help path detection on cluttered environments. The combination of the 3D terrain map with a terrain classification increases the reliability of the traversability map. This facilitates the segmentation of traversable region by applying a watershed on the traversable cost map. This segmentation is used then as a ground-true label for feeding back the classifier; even though the resulting path is not actually a ground-true, the estimation allows getting actual training examples to improve the classifier model and deal with changing environments.

ACKNOWLEDGMENT

The authors would like to thank São Paulo Research Foundation (FAPESP), Grant #2012/21079-1, for its financial support.

REFERENCES

- [1] A. Cappalunga, S. Cattani, A. Broggi, M. S. McDaniel, and S. Dutta, "Real time 3D terrain elevation mapping using ants Optimization Algorithm and stereo vision," *2010 IEEE Intelligent Vehicles Symposium*, pp. 902–909, Jun. 2010.
- [2] A. L. Rankin, A. Huertas, and L. H. Matthies, "Stereo-vision-based terrain mapping for off-road autonomous navigation," *Proceedings of SPIE*, vol. 7332, pp. 733 210–733 210–17, 2009.
- [3] U. Malik, S. Ahmed, and F. Kunwar, "A self-organizing neural scheme for road detection in varied environments," *Neural Networks (IJCNN)*, pp. 3049–3054, 2011.
- [4] S. Zhou, J. Gong, G. Xiong, H. Chen, and K. Iagnemma, "Road detection using support vector machine based on online learning and evaluation," in *Intelligent Vehicles Symposium (IV)*, 2010 IEEE. IEEE, 2010, pp. 256–261.
- [5] S. Zhou, J. Xi, P. Salesses, and K. Iagnemma, "Self-Supervised Learning to Visually Detect Terrain Surfaces for Autonomous Robots Operating in Forested Terrain," *Journal of Field Robotics*, 2012.
- [6] R. Hadsell and P. Sermanet, "Online learning for offroad robots: Using spatial label propagation to learn long-range traversability," *Proc. of Robotics*, 2007.
- [7] A. Broggi, E. Cardarelli, S. Cattani, and M. Sabbatelli, "Terrain Mapping for Off-road Autonomous Ground Vehicles Using Rational B-Spline Surfaces and Stereo Vision," *IEEE Intelligent Vehicles Symposium 2013*, 2013.
- [8] M. Tuceryan and A. Jain, "Texture Analysis," in *The Handbook of Pattern Recognition and Computer Vision*, 2nd ed., C. Chen and L. Pau, Eds. World Scientific Publishing Co., 1998, pp. 1–41.
- [9] C. Hsu, C. Chang, C. Lin, and Others, "A practical guide to support vector classification," Department of Computer Science National Taiwan University, Tech. Rep. 1, 2010.
- [10] N. Bova, S. Cattani, F. Sassi, S. Cagnoni, and A. Broggi, "Unpublished master's thesis. Artificial vision and soft computing techniques for viable space estimation in the automotive field," p. 124, 2009.