Boosting Support Vector Machines for RGB-D Based Terrain Classification

Submitted: 6th June 2014; accepted: 15th July 2014

Jan Wietrzykowski, Dominik Belter

DOI: 10.14313/JAMRIS_3-2014/24

Abstract:
This paper deals with the terrain classification problem for an autonomous mobile robot. The robot is designed to operate in an outdoor environment. The classifier integrates data from RGB camera and 2D laser scanner. The camera provides information about visual appearance of the objects in front of the robot. The laser scanner provides data about distance to the objects and their ability to reflect infrared beam. In this paper we present the method which create terrain segments and classifies them using joint application of Support Vector Machine (SVM) classifier and AdaBoost algorithm. The classification results of the experimental verification are provided in the paper.

Keywords: *Terrain classification, mobile robot, RGB-D*

1. Introduction

Autonomous navigation in urban environment is a challenge for mobile robots. The robot which operates in urban space should localize itself, find the path to the goal position and avoid obstacles. Moreover, it should obey rules which are designed for humans. It is obvious that autonomous car-like robot should follow the road. Access to the pavement is prohibited. The robot which is designed as short distance courier should use pavement for locomotion and avoid road as possibly dangerous area. The access to the lawn should also be prohibited. In this case it isn't dangerous for the robot, but such a behavior is against principles of community life. To obey all rules the robot should recognize various terrain types.

The autonomous operation in urban environment differs to operation in off-road natural environment. The first difference is related to traversability assessment methods. Outdoor and and off-road locomotion takes into account mainly the shape of the terrain. The terrain type does not play an important role. Grass as well as asphalt is considered as traversable. Such situation is not acceptable in urban environment. Moreover, in off-road environment the borders between various regions are difficult do distinguish (e.g. the grass can be also found on the field track). In man-made environment most of objects and terrain types have standard size, color and location. On the other hand robot which operates in urban-like environment has to distinguish between very similar areas like road and pavement.

1.1. Problem statement

Our goal is to create the robot which can navigate in urban environment. The paper is focused on terrain classification which is important part of the navigation system of a mobile robot. We are interested in robotic competitions for delivery or search and rescue. The scenario of such challenges include autonomous navigation on paved park roads (Robotour) or finding and fetching an object (e.g. 1 kilo "bag of gold" in Robots Intellect competition).

The robot which navigates in urban environment can't use only depth sensors to create environment model. Some obstacles, however flat, are not traversable (e.g. lawn). Other places like pedestrian crossing should be recognized to apply special strategy for traversing. This can be done by visual camera. Using monocular RGB camera only the robot would have problems to distinguish between asphalt and flat, vertical and gray wall. It is much easier to classify terrain using two complementary sensors.

In the paper we present way to classify terrain using data from RGB-D sensors (in our case laser scanner and visual camera). We are interested in segmentation of an image and labeling detected areas. To this end, we applied classification strategy which utilizes SVM classification and AdaBoosting. We present results from indoor and outdoor experiments. The obtained results are compared with other approaches to show efficiency of the proposed method.

1.2. Related work and research contribution

Most of the existing terrain classification methods employ RGB cameras for features extraction [5, 13]. In our work we increase the robustness of the classification procedure by incorporation information about depth of the scene. Laible et al. presented that the classification accuracy can be increased by the analysis of the whole scene and taking into account neighboring regions [15]. The decision about terrain type can't be taken using local terrain properties only. Context, neighboring terrain types and location of the considered image segment play important role in the classification procedure. To join information from weak classifiers Laible et al. proposed the application of Conditional Random Fields [15].

The joint application of 2D laser scanner and RGB camera to terrain classification is not new. Dahlkamp et al. proposed to use data from range finder to supervise learning algorithm [6]. The surface model obtained from depth data is used to find a traversable area (road). Then, the visual data from a camera is used by a learning algorithm. The classification method uses mixture of Gaussians in RGB space to classify the terrain. The model is updated whenever new learning dataset is provided by self-supervising procedure. The re-learning procedure allows the system to adapt even when the road changes from gray asphalt to green grass. In our case this situation is not desirable. The classifier should determine not only the traversability, but also the terrain type. The grass (however flat) should be also considered as an obstacle for our robot. In contrast to method presented by Dahlkamp et al. we use RGB and depth data during classification stage.

A reliable terrain classification can be also obtained using visual features and SVM classification [9]. In the method proposed by Filitchkin and Byl SURF features are used. To deal with various surfaces, which differ with number of visual features, the optimization on Hessian threshold detection is proposed. However, the feature-based classification is sensitive to motion blurring problem. Thus, we decided to use few independent sources of information.

Most reliable classification methods suffer from high computational cost. Angelova et al. proposed a cascade of classifiers instead of single, multi-dimensional classification to obtain high speed and preserve high classification accuracy [1]. They take advantage of the fact that some terrain types might be easily separated from the others. This observation can be used to create decision tree. The classification starts from the fastest classification sub-procedure. The most computationally expensive procedures are performed at the end an only for regions which are difficult to distinguish.

Additional classification capabilities are available for legged robots. Such robots can use force/torque sensors as an additional source of information to classification procedure [12, 19]. Also wheeled robots can use properties of the contact with the ground to support classification procedure (e.g. vibrations which propagate through suspension structure [11]).

2. Perception and data acquisition

The environment perception is based on two sensor: a generic USB camera (Microsoft LifeCam Studio) and laser range finder. The robot acquires VGA (640×480) images. VGA resolution is sufficient for classification and allows to decrease the computation time of the procedure. The Hokuyo UTM-30LX laser finder used in this research can operate outdoor. The range of the sensor is up to 30 m. The angular resolution is 0.25° and each scan takes 25 ms. Single scan contains information about terrain profile. When the robot moves forward or rotates it acquires 3D shape of the environment. Both sensors are tilted down to acquire terrain properties.

To create map of the environment the robot has to determine the position of the sensors in global coordinate system O_G at each scan of range finder. The robot uses GPS, encoders and Inertial Measurement Unit (IMU) to localize itself. Data from all sensors are integrated using Kalman Filter. The robot is equipped with IMU CHR-UM6 sensor on board. It allows to measure properly the shape of the environment on rough terrain. The robot can take into account the inclination of the platform during integration of the measurements. In our research we use two various mobile platforms. However, the presented classification method is platform independent.

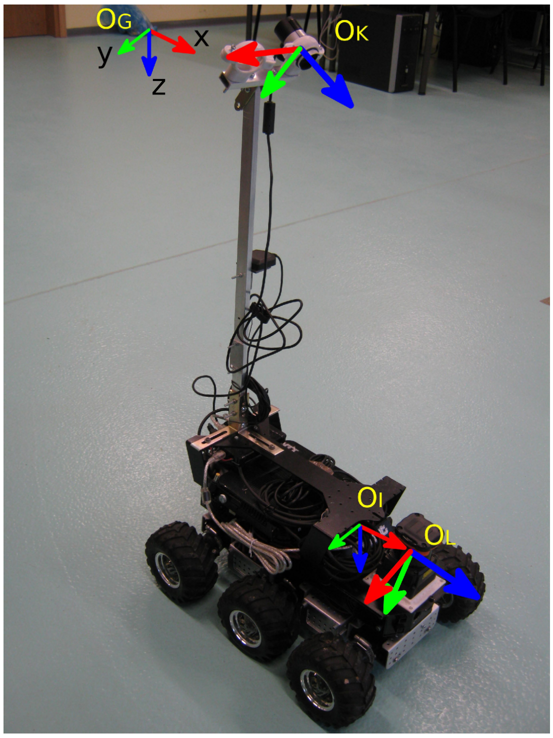

Fig. 1. Configuration of the sensors attached to the robot's platform

The configuration of sensors is presented in Fig. 1. The coordinate systems O_K, O_I and O_L are attached to the camera, IMU and laser range finder, respectively. To integrate data from all sensors the correspondence between each pixel of the camera image and points of the laser scan has to be known. To this end, the pose of each sensor has to be determined by the calibration procedure. The calibration procedure also determines intrinsic parameters of the camera. We applied Camera Calibration Toolbox for Matlab by Jean-Yves Bouguet [4] to find focal length and location of the principal point.

To find relation between camera and laser range finder a plane-to-line fitting method was applied [20]. We can use checkerboard marker from intrinsic calibration to determine a plane which represents marker. From the laser scan we can find an equation of the line located on on this plane. From the measurements set we can compute transformation between camera and laser scanner coordinate system. To present calibration results between camera and laser scanner we draw single scan on the camera image. The result is presented in Fig. 2.

Moreover, we should find orientation between exteroceptive sensors (camera and LRF) and IMU unit. To this end, we used the method proposed by Lobo

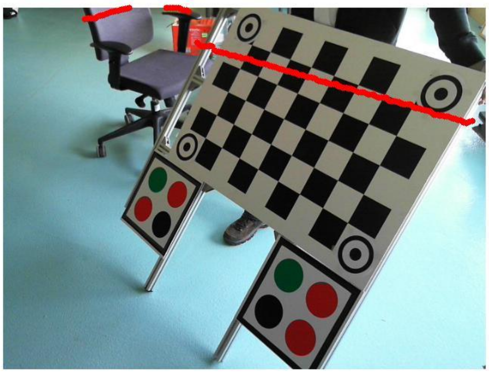

Fig. 2. Calibration results of the camera and the laser range finder

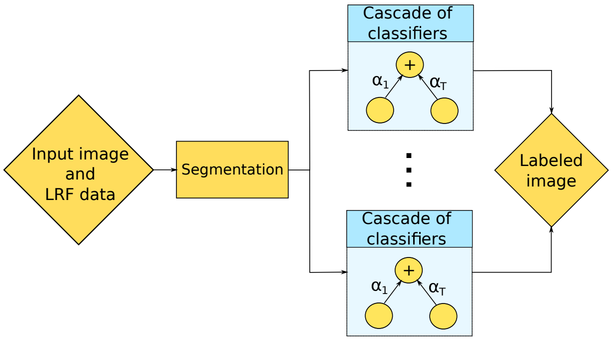

Fig. 3. Structure of the terrain classification procedure which uses RGB-D input data and returns labeled image

et al. [16]. In this case we use checkerboard marker which is perpendicular to gravity vector. Taking into account orientation measured by the camera we can find orientation offset of the IMU unit.

Finally to compute pose of each measured point P_L in global coordinate system O_G we apply (1):

$$P_G = {}^{G}A_I \cdot {}^{I}A_K \cdot {}^{K}A_L \cdot P_L, \quad (1)$$

where ${}^{K}A_L$ is a transformation from the camera coordinate system to the laser coordinate system, ${}^{I}A_K$ is a transformation from IMU unit to the camera coordinate system and ${}^{G}A_I$ is the IMU unit pose in the global coordinate system obtained from the localization system.

3. Terrain classification

The input to our system is data from RGB camera and Hokuyo laser scanner. The architecture of the classification procedure is presented in Fig. 3. At the beginning the segmentation is performed using RGB image. For each segment we compute k visual and depth features $(f_1, ..., f_k)$. A set of features is then used for classification. We decided to use combination of SVM weak classifiers and boosting technique to join results (Fig. 4). It was shown that this approach has better performance in training time [10]. We also show that classification results are better. To use boosting technique we should use weak classifiers (which perform

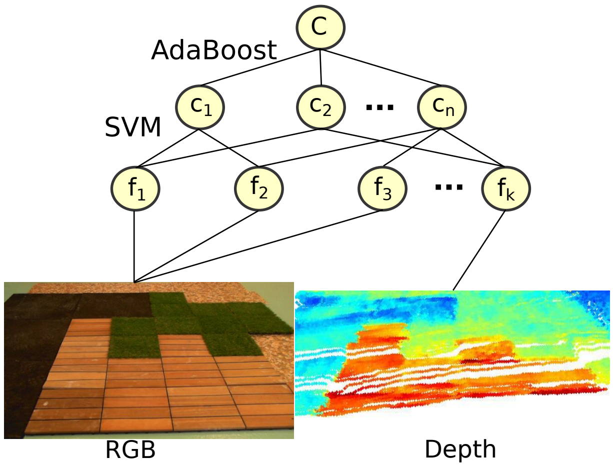

Fig. 4. Classification scheme with SVM weak classifiers and AdaBoost

better than random classifier, e.g. Decision Stump). Instead we can use strong classifiers e.g. SVM or Neural Network which are appropriately weakened [7]. In our method n weak SVM classifiers (c_1,...,c_n) are used. The output C from the classifier is the terrain category recognized by the system.

3.1. Segmentation

In our method we perform image segmentation and then classification for each RGB-D segment. We avoid classification for each pixel separately because we don't always have corresponding depth for each pixel. Moreover, single pixel does not contain all information about the terrain properties like roughness obtained from depth data or variance of color. We also avoid dividing the image into regular mesh [14, 17]. Constant and rectangular region may contain two separate terrains and such a cell should be classified as two separate classes. Instead, we perform image segmentation and then we classify each region separately.

We use a method proposed by Pedro Fenzenszwalb and Daniel Huttenlocher for image segmentation [8]. The segmentation method used in our system divides an image into components. The behavior of the segmentation method is specified only by two parameters. First parameter k_c is responsible for preferred component size. Second parameter S_{min} represents the minimal size of components.

Before segmentation the image is smoothed by Gaussian filter with $\sigma = 0.8$. At the beginning of the segmentation the algorithm creates initial graph G (Fig. 5). Each edge E of the graph connects two vertices, representing single neighboring pixels of the image. The weight w of an edge is computed using difference in pixel color values [r,g,b] with an Euclidean distance. Then, edges of the graph are sorted in an ascending order with respect to weights w. A set of segments is initialized – each vertex of the graph represent separate segment. Next, for each edge which belongs to separate components weight w_q is computed. The computed weight w_q is compared with threshold

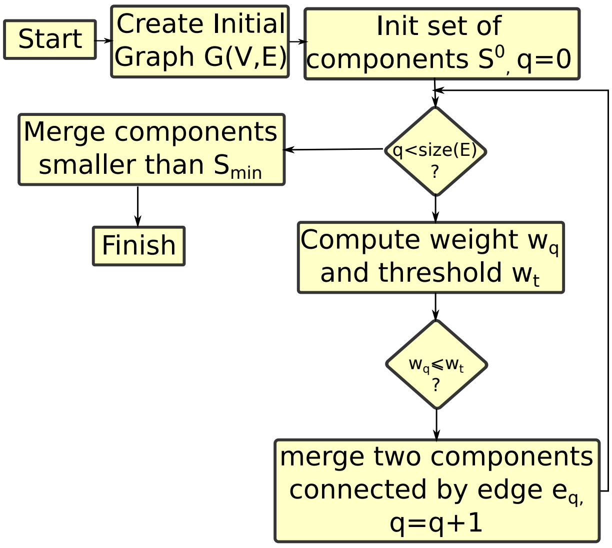

Fig. 5. Segmentation procedure used for image partitioning

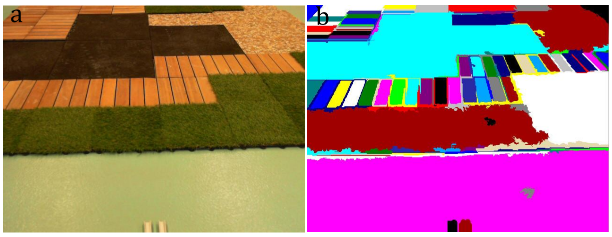

Fig. 6. Segmentation results – original image (a) and output components (b)

weight w_t:

$$w_t = \min(INT(v_i) + \frac{k_c}{Size(v_i)}, INT(v_j) + \frac{k_c}{Size(v_j)}), \quad (2)$$

where v_i is the considered vertex, v_j is the neighboring vertex, $Size(v)$ is the size of component represented by vertex v and $INT(v)$ is the maximal weight between vertices which create the whole component. If w_q is smaller or equal w_t vertices are merged into single component S. In an opposite case, the configuration of segments does not change. Finally, the algorithm removes components which are smaller than threshold S_{min}. Components which are too small are merged with neighboring components. The algorithm returns a set of components S.

The results of the segmentation procedure are presented in Fig. 6.

3.2. Classification

For classification purpose we use Support Vector Machine supervised learning algorithm [3]. We decided to use SVM because it works well with multidimensional input vector. The output from the classifier is the value of assignment to each category of terrains. We created five weak classifiers. The input for each classifier is defined as follows:

1) two dimensional histogram of values in *Hue-Saturation* color space (4×4 bins) converted to one-dimensional vector,
2) 8 bin histogram of *Value* in HSV space,
3) mean and covariance matrix for pixels in HSV space converted to a 1×12 vector,
4) mean and covariance matrix for values of depth and intensity from Hokuyo laser scanner converted to a 1×6 vector,
5) 25 bin histogram of intensity values from Hokuyo laser scanner.

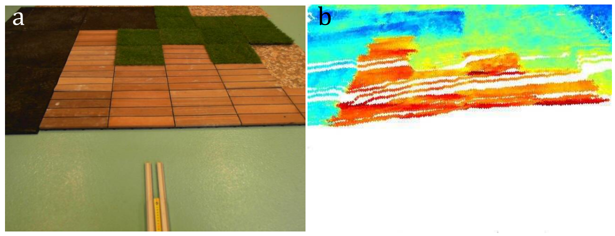

Fig. 7. Intensity values obtained from Hokuyo laser range finder – observed scene (a) and registered intensity values (b)

First three classifiers use color image feature as an input for classification. The next two classifiers use data from depth sensor. We use depth data directly as well as intensity values which are provided by the Hokuyo driver. The intensity value depends on the color and texture of the surface. Thus, intensity value provides important information about observed surface [15]. Example intensity values for the various terrain types are presented in Fig. 7. Using intensity values we can easily distinguish between various types of terrain without direct information about color of the surface.

For boosting we use improved version of AdaBoost algorithm [18] based on MultiBoost implementation [2] which deals with multi-class weak-learning.

4. Results

First experiments were performed indoor. Our goal was to avoid problems with uncertainty of the localization system which introduces mapping error. We created mockup with various terrain types like artificial grass, elastic gum, timber and tile floor (Fig. 7a).

For training classifiers we used 33 manually marked scenes. Next 35 scenes were used for testing. For segmentation we set $k = 50$ which allows to obtain 3000 training samples. For testing we use $k = 200$ which allows to obtain segments which represent bigger area. Thus, we avoid situations when grass is divided into green patches representing grass and small black patches representing soil. We are interested in classification of the whole region with heterogeneous texture.

Example classification results are presented in Fig. 8. Colors in Fig. 8b represent various type of terrain: green – grass, brown – timber floor, blue – asphalt, yellow – rocky terrain. Only some small areas are classified improperly. The component of the timber floor is classified as a rocky terrain. Also small re-

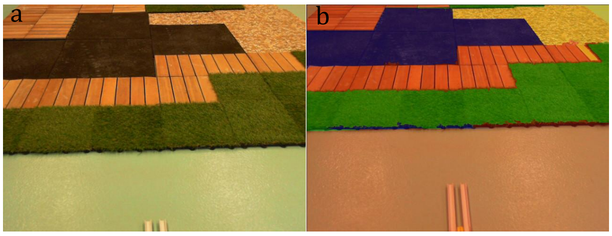

Fig. 8. Results – the example scene (a) and classification results

gion between the grass and the floor is classified improperly as an asphalt (mainly because of black color of this part).

Tab. 1. Confusion matrix for indoor experiment

terrain type	grass	timber	rocks	asphalt
grass	94%	5%	1%	0%
timber	0%	90%	10%	0%
rocks	0%	3%	96%	1%
asphalt	0%	1%	2%	96%

We also performed statistical analysis for the whole testing set. The results are presented in Tab. 1. Each row in the table represent the terrain type marked by the expert. Each column represent output from proposed classification method. The classification results p_{c_1,c_2} presented in Tab. 1 are computed as follows:

$$p_{c_1,c_2} = \frac{N_{c_1}}{N_{c_2}} \cdot 100\%, \tag{3}$$

where N_{c_1} is the number of pixels classified as class c_1 and N_{c_2} is the number of pixels marked by expert as class c_2. It means that 94% of pixels which belong to grass are classified properly as a grass, 5% of pixels are classified as timber and 1% as rocks.

Tab. 2. Comparison between various configurations of the proposed classifier and input features

	classificatory type				
terrain type	C_c	C_l	MON	C_{one}	C_{kl}
grass	43%	95%	86%	95%	94%
timber	94%	16%	75%	71%	90%
rocks	39%	85%	98%	98%	96%
asphalt	64%	97%	96%	89%	96%
average	60%	75%	86%	89%	94%

We also compared various configurations of classifiers and input features. We compared six configurations:

1) C_c – SVM classification with AdaBoost and features computed using RGB image only
2) C_l – SVM classification with AdaBoost and features computed using depth data only
3) MON – SVM classification without AdaBoost using single vector of features computed using RGB and depth data
4) C_{one} – SVM classification with AdaBoost and single features vector computed using RGB and depth data
5) C_{kl} – SVM classification with AdaBoost and features computed using RGB and depth data (solution proposed in the paper)

The results of the comparison experiment are presented in Tab. 2. The best performance is obtained by classificator proposed in the paper. The average classification accuracy is 94% while the performance for standard SVM classificator is 86%.

Tab. 3. Computation time

task		time [s]
segmentation	sorting	0,699
	segmentation	0,414
	merging	0,274
features extraction	pre-processing	0,122
	computation	0,022
classification	computation	0,260
total		1,791

We also checked the computation time of each element of the proposed procedure. The results are presented in Tab. 3. The most consuming part is the segmentation procedure. Sorting of edges takes 0.7 s, segmentation 0.4 s and removing segments smaller than threshold takes almost 0.3 s. Features extraction is faster and takes only 0.144 s including preparation of depth and color data and features computation. The classification takes 0.26 s. The whole classification procedure takes 1.791 s. It is fast enough to implement the method on the real robot because the robot needs at least 2 s to acquire information about new terrain.

4.1. Outdoor experiment

We also performed outdoor experiment on the robot with the final setup of sensors. The robot classifies grass, asphalt and two types of pavements (pave1 and pave2 in Tab. 4). The color and the geometrical properties of the pavements and the asphalt are similar. Thus we added a new set of features which allows to distinguish between similar terrain types. The new inputs of the classifier are related to shape of the segmented regions. For each region we detect line segments using RANSAC. The line segments are used to compute additional input features. The input values are as follows:

1) regularity coefficient which is computed as a sum of line segments lengths divided by the total number of pixels which belong to border of the region,
2) mean of line segments lengths,
3) variance of line segments lengths,
4) number of line segments,
5) 10 bin histogram of line segments orientations.

The classification results are presented in Tab. 4. The average classification precision is 82%. It is significantly smaller in comparison to results of the experiments performed indoor. The outdoor experiments on

Tab. 4. Confusion matrix for an outdoor experiment

terrain type	grass	pave1	pave2	asphalt
grass	99%	1%	1%	0%
pave1	18%	81%	2%	0%
pave2	4%	25%	68%	3%
asphalt	0%	3%	37%	60%

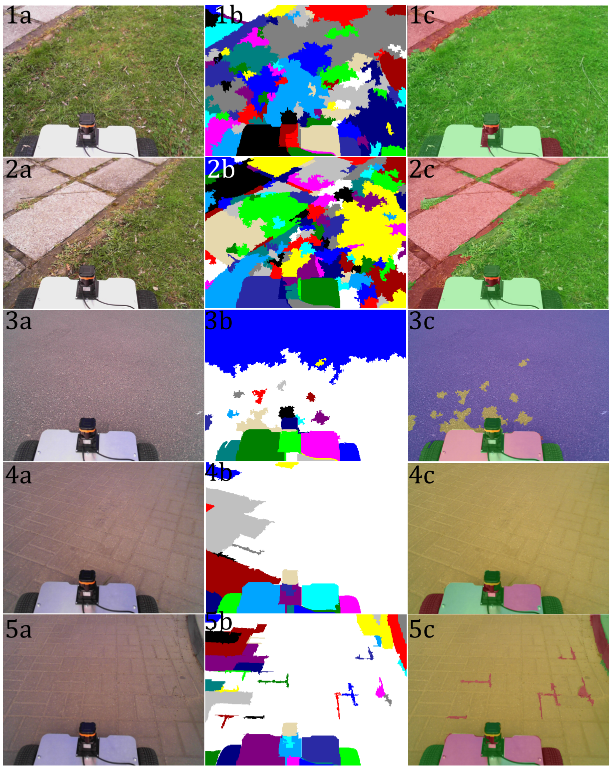

Fig. 9. Results of the outdoor experiment – the example scene (a) segmentation (b) and classification (c) results.

the real robot are more challenging. The first difficulty is connected to similarity between classified regions. The other difficulties are caused by imprecise localization system (odometry and IMU). The robot moves in irregular terrain. Thus, the imprecise measurements of the inclination of the robot's platform causes incorrect location of the 3D points obtained from range measurements.

The example classification results are presented in Fig. 9. Colors in Fig. 9c represent various type of terrain: green – grass, yellow – pavement 1, red – pavement 2, blue – asphalt. The classification results are accurate enough to use the proposed method on the real robot dedicated to robotic competition.

5. Conclusions and future work

In the paper we presented the terrain classification method for the mobile robot. We show that the performance of the classification can be increased by using boosting technique to combine output from weak SVM classifiers. The results for SVM and AdaBoost classifier are better than for single SVM classifier with multidimensional features vector. SVM algorithm works efficiently with multi-dimensional problems. By using our method we reduce the dimensionality of the problem. The performance of the classification increases as a result.

To show performance and advantages of our method we performed experiments indoor on terrain mockup and outdoor in real environment. We carried out the analysis of classification results. We compared various combinations of classification input and configurations of the classifier. We conclude that the classification results are better when depth data are used. The advantages of the method which uses the data from LRF are mainly due to the intensity values. They provide information about properties of the object's surface which is well utilized by the classifier.

We also show the computation time of each element of the procedure. The most expensive part is segmentation. It takes more than 1 s to divide the image into segments. From the application point of view we are going to replace existing procedure by the faster one. On the other hand our goal is to increase performance of the segmentation procedure. To this end, we are going to use methods which take into account depth and color data simultaneously during segmentation.

In future we are going to add next layer to classification method. Our goal is to take into account classification results of neighboring segments as well as depth and color of the considered segment. We believe that context-aware segmentation will bring better efficiency of the classification procedure.

AUTHORS

Jan Wietrzykowski – Poznań University of Technology, Institute of Control and Information Engineering, ul. Piotrowo 3A, 60-965 Poznań, Poland, e-mail: Jan.Wietrzykowski@student.put.poznan.pl.

Dominik Belter* – Poznań University of Technology, Institute of Control and Information Engineering, ul. Piotrowo 3A, 60-965 Poznań, Poland, e-mail: Dominik.Belter@put.poznan.pl.

*Corresponding author

REFERENCES

[1] A. Angelowa, L. Matthies, D. Helmick, and P. Perona, "Fast terrain classification using variable-length representation for autonomous navigation". In: *Proceedings of the conference on Computer Vision and Pattern Recognition*, Minneapolis, USA, 2007, pp. 1–8, doi: 10.1109/CVPR.2007.383024.

[2] D. Benbouzid, R. Busa-Fekete, N. Casagrande, F.-D. Collin, and B. Kegl, "Multiboost: a multi-purpose boosting package", *Journal of Machine Learning Research*, vol. 13, 2012, pp. 549–553.

[3] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[4] J.-Y. Bouguet. "Camera calibration toolbox for matlab", 2014, www.vision.caltech.edu/bouguetj/calib_doc.

[5] J. Chetan, M. Krishna, and C. Jawahar, "Fast and spatially-smooth terrain classification using monocular camera". In: *Proceedings of 2010 20th International Conference on Pattern Recognition*, Istanbul, Turkey, 2010, pp. 4060–4063.

[6] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski, "Self-supervised monocular road detection in desert terrain". In: *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, 2006.

[7] T. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization", *Machine Learning*, vol. 40, no. 2, 2000, pp. 139–157.

[8] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation", *International Journal of Computer Vision*, vol. 59, no. 2, 2004, pp. 167–181, doi: 10.1023/B:VISI.0000022288.19776.77.

[9] P. Filitchkin and K. Byl, "Feature-based terrain classification for littledog". In: *Proceedings of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura, Portugal, 2012, pp. 1387–1392, doi: 10.1109/IROS.2012.6386042.

[10] E. García and F. Lozano, "Boosting support vector machines". In: *Proceedings of 5th International Conference on Machine Learning and Data Mining in Pattern Recognition*, Leipzig, Germany, 2007, pp. 153–167.

[11] I. Halatci, C. Brooks, and K. Iagnemma, "Terrain classification and classifier fusion for planetary exploration rovers". In: *Proceedings of 2007 IEEE Aerospace Conference*, Big Sky, USA, 2007, pp. 1–11, doi: 10.1109/AERO.2007.352692.

[12] M. Hoepflinger, C. Remy, M. Hutter, L. Spinello, and R. Siegwart, "Haptic terrain classification for legged robots". In: *Proceedings of 2010 IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, USA, 2010, pp. 2828–2833, doi: 10.1109/ROBOT.2010.5509309.

[13] R. Karlsen and G. Witus, "Terrain understanding for robot navigation". In: *Proceedings of 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Diego, USA, 2007, pp. 895–900, doi: 10.1109/IROS.2007.4399223.

[14] Y. Khan, P. Komma, and A. Zell, "High resolution visual terrain classification for outdoor robots". In: *Proceedings of IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, Barcelona, Spain, 2011, pp. 1014–1021, doi: 10.1109/ICCVW.2011.6130362.

[15] S. Laible, Y. Khan, and A. Zell, "Terrain classification with conditional random fields on fused 3d lidar and camera data". In: *Proceedings of European Conference on Mobile Robots*, Barcelona, Spain, 2013, pp. 172–177, doi: 10.1109/ECMR.2013.6698838.

[16] J. Lobo and J. Dias, "Relative pose calibration between visual and inertial sensors", *International Journal of Robotics Research*, vol. 26, no. 6, 2004, pp. 561–575, doi: 10.1177/0278364907079276.

[17] D. Maier, C. Stachniss, and M. Bennewitz, "Vision-based humanoid navigation using self-supervised obstacle detection", *International Journal of Humanoid Robotics*, vol. 10, no. 2, 2013, doi: 10.1142/S0219843613500163.

[18] R. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions", *Machine Learning*, vol. 37, 1999, pp. 297–336, doi: 10.1145/279943.279960.

[19] K. Walas, A. Schmidt, M. Kraft, and M. Fularz, "Hardware implementation of ground classification for a walking robot". In: *Proceedings of the 9th International Workshop on Robot Motion and Control*, Wąsowo, Poland, 2013, pp. 110–115, doi: 10.1109/RoMoCo.2013.6614594.

[20] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004, pp. 2301–2306, doi: 10.1016/j.proeng.2012.01.669.