EAN 2017
55th conference on experimental stress analysis

Improvements of VFH* Method

Andrej Babinec[1,a], Marian Kľúčik[1,b] and Zuzana Mikulová[1,c]
[1]FEI STU, Ilkovičiva 3, 812 19 Bratislava, Slovakia
[a]andrej.babinec@stuba.sk, [b]marian.klucik@stuba.sk, [c]zuzana.mikulova@stuba.sk

Keywords: Mobile robot, navigation, moving obstacles.

Introduction

A mobile robot operating in unknown environment should have got a reliable control system [1] with implementation of a reactive navigation as well as a path planning. The reactive navigation is necessary because it allows the robot to avoid unknown obstacles and therefore it protects the robot and environment against any mechanical damages [2,3]. Forasmuch as current robots have modern sensors on board, e.g. laser rangefinders or visual systems, they have wider knowledge about the larger surrounding. This is why the local forms of path planning have to be employed. It makes the robot motion more fluent and produces shorter trajectories.

There are many algorithms focused on reactive navigation and path planning. In particular, Vector Field Histogram methods [4,5,6] are unique because they are very effective in obstacle avoidance and they are easily customizable. Even more, VFH* method yet combines reactive navigation with local path planning into one complex algorithm. Unfortunately, these algorithms are natively designed for static environment so moving objects are in most cases impassable obstacles.

Further in the article, a new reactive navigation of mobile robot based on VFH* method is presented. The algorithm is modified so that the robot is able to plan a local avoidance maneuver even in the presence of moving obstacles. The modification consists of several improvements, which are described in following section.

Modifications in the Method

Considering the non-static environment it was necessary to design how to include the information about mobile obstacles into the algorithm’s computation. Recognition of moving objects is discussed in many papers [7,8,9]. However, for the purpose of reactive navigation it is necessary to know only a final representation of moving objects, because they enter the decision-making process. So we need to create some form to store information about the obstacles suitable for further processing during local path planning. Details are presented in first subsection.

The original VFH* method includes VFH+ method, which is its essential component. In VFH+ method, polar histograms are created from local grid map. The histograms consist of n bins. Each bin corresponds to particular angle sector coming out from the robot position. The near space around the robot (active window) is therefore divided into n angle sectors. Each histogram bin carries the value proportional to the amount of occupied cells in that direction. The potential passages are identified in valleys of the primary histogram. Then the single one passage from the set of identified passages is chosen that leads the robot to the goal along the smooth and safe trajectory. For the purpose of taking a robot size into account and for simplifying it to a point, the occupied cells in a grid map are enlarged by the circle with radius r_{r+s} and the enlargement angle $\gamma_{i,j}$ is defined by Eq. 1 [5]. For each obstacle cell in the map, the angle $\gamma_{i,j}$ designates the membership of the sector k by specific rules. These rules had to be modified in order to adapt the method to dynamic environment. Second subsection of this section is focused on this modification in geometrical constrains and its consequences.

$$\gamma_{i,j} = \arcsin \frac{r_{r+s}}{d_{i,j}} \tag{1}$$

The VFH* method produces look-ahead tree through the free space in near robot surrounding based on potential passages given by VFH+. The optimal branch in the look-ahead tree predicts the future change of the robot’s direction. As this change is known at the end of the tree development stage, it is convenient to process it and properly deflect the robot's next direction. These improvements are contained in third subsection.

Representation of Mobile Obstacle. In histogramic algorithms, static objects are stored in a grid map that is given as an input to the navigation algorithm. Besides, moving objects have to be included. While the laser rangefinder is utilized in many robotic systems, it is convenient to use the laser data directly in moving objects representation. The output of the rangefinder is an array of couples (φ_i, d_i). It describes the sensed distance to an obstacle in a particular direction. Moving obstacle can be detected after an appropriate filtration and segmentation. Then the set of distances measured towards the moving obstacle from the point where the obstacle was recognised for the first time is included to a data structure in Eq. 2 along with the robot position and the recognized velocity vector of the obstacle. The symbols are explained by the Fig. 1, where x^R, y^R, φ^R denote the robot position where the obstacle was detected for the first time, k is the index of the first laser ray that belongs to the obstacle and n is the number of the laser rays that belong to the obstacle.

$$O^m = \{x^R, y^R, \varphi^R, \{d_k,...,d_{k+n}\}, k, n, v_D, \varphi_D\}. \tag{2}$$

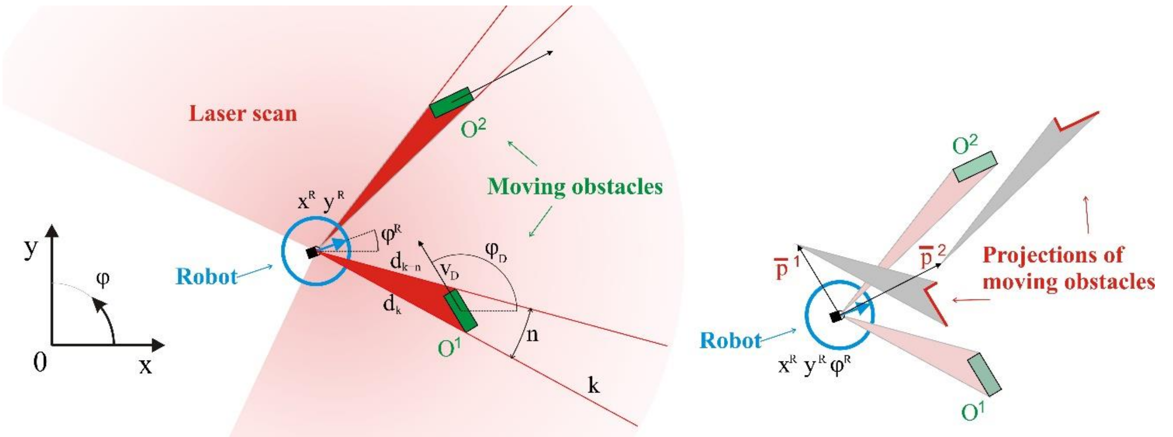


Fig. 1 The laser scanner segments which serve as the representations of moving obstacles, and their example of projections.

This representation allows preserving the original object shape. The objects are inserted to the grid map just when they have to be presented during a computation. After the processing, the objects are removed from the grid.

This approach allows to simulate and predict a motion of the moving obstacle O^m by shifting the reference point (x^R, y^R) along the projection vector $\vec{p}^m$, which goes out from the reference point in the direction of obstacle motion φ_D. Therefore, it is necessary to take into consideration also a time difference between current state and predicted state. Inclusion of this time difference into the computation of the look-ahead tree is crucial improvement of the method VFH*, which allows employment of the method into the non-static environment.

Improvement in Geometrical Constraints. A definition of so called enlargement angle in original VFH* does not count with the possibility that the robot and obstacle could intersect and it is valid only for $d_{i,j} \geq r_{r+s}$. In the virtual scene, when the algorithm predicts motion of mobile obstacles it is possible that predicted projection of the obstacle hit the projection of the robot. Such a node in look-ahead tree that leads to the collision has to be closed so the optimal branch of the tree should be allowed to grow through other passages in free space. It is necessary that these virtual collisions are mathematically possible in order to be recognised. Therefore, the definition of the enlargement angle was extended, which can be seen in Eq. 3.

$$\gamma_{i,j} = \begin{cases} \arcsin \dfrac{r_{r+s}}{d_{i,j}} & if \quad d_{i,j} \geq r_{r+s} \\ \dfrac{\pi}{2} & if \quad r_r < d_{i,j} < r_{r+s} \\ \pi & if \quad d_{i,j} \leq r_r \end{cases} \tag{3}$$

Improvement of Final Decision Step. During the stage of look-ahead tree development, a direction θ_i, in which the robot projection came from the node i-1 to node i, is stored. After determination of optimal branch, directions θ_i of the branch are compared with each other. The direction that is significantly different from current robot orientation φ^R, affects a future robot's motion mostly. The node i with maximal δ_i according Eq. 4 is called a deflecting node. Then the direction from the root node in position (x^R,y^R) to the deflecting node is called a deflecting direction θ_d (see Fig. 2).

$$\delta_i = \left|\varphi^R - \theta_i\right| \tag{4}$$

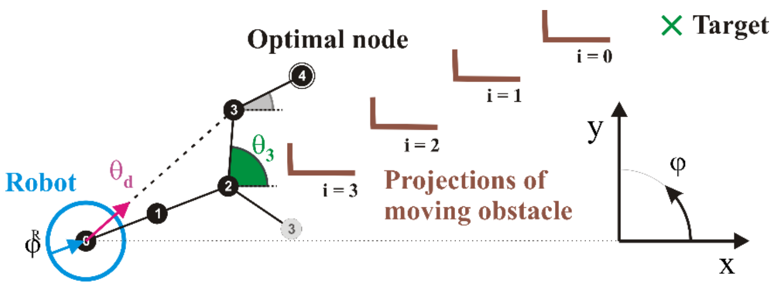


Fig. 2 Determination of the deflecting direction θ_d.

If the deflecting direction respects the restrictions in the primary node (root) given by the histograms of VFH+ method, it becomes a desired direction for the next robot step. If θ_d lies outside of optimal opening, another direction is used according the Fig. 3.

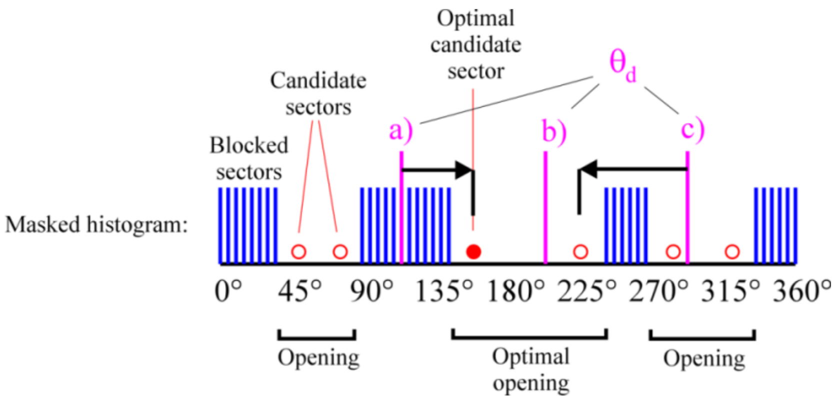


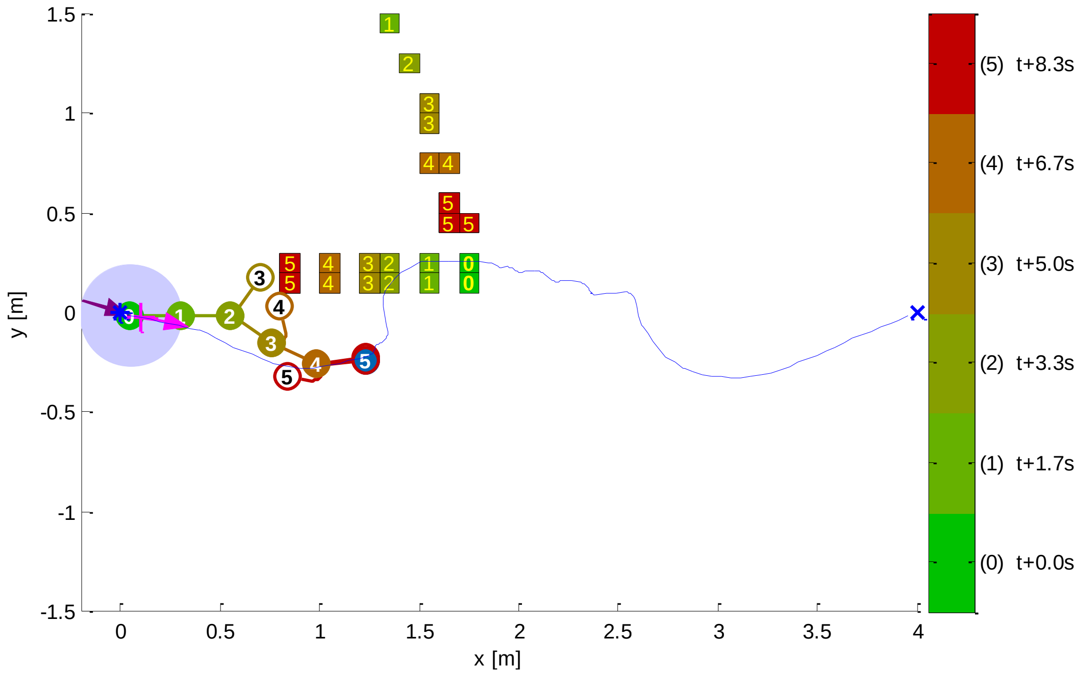
Fig. 3 Restrictions for the deflecting direction given by VFH+ method. θ_d lies: a) in blocked sector – the nearest candidate sector of the optimal opening is used; b) in optimal opening between its marginal candidate sectors – θ_d is used; c) in another opening – the nearest candidate sector of the optimal opening is used.

Verification

The proposed improved method was implemented to our indoor differentially driven mobile robot, which was equipped with laser scanner. During experiments all process data had been recorded in order to be visualized. In following figures, light blue large circle represents the robot and blue curve is its overall recorded path from start (asterisk) to goal (cross). Moving obstacles are depicted by colored square cells of the grid map. Green cells numbered by zero represent the actual position of the obstacles in time *t* along with actual robot position in the plane. Other colored squared cells represent projected positions of the obstacles into future using respective time differences. These projections are utilized during look-ahead tree development. In each node of the tree (small circles), there are only corresponding projections of the obstacles (with the same number and color) taken into computation of next node. Blue node is the optimal node on the optimal branch. Deep-violet arrow means actual robot direction φ^R. Light-violet arrow is the deflected direction θ_d after consideration of all restrictions.

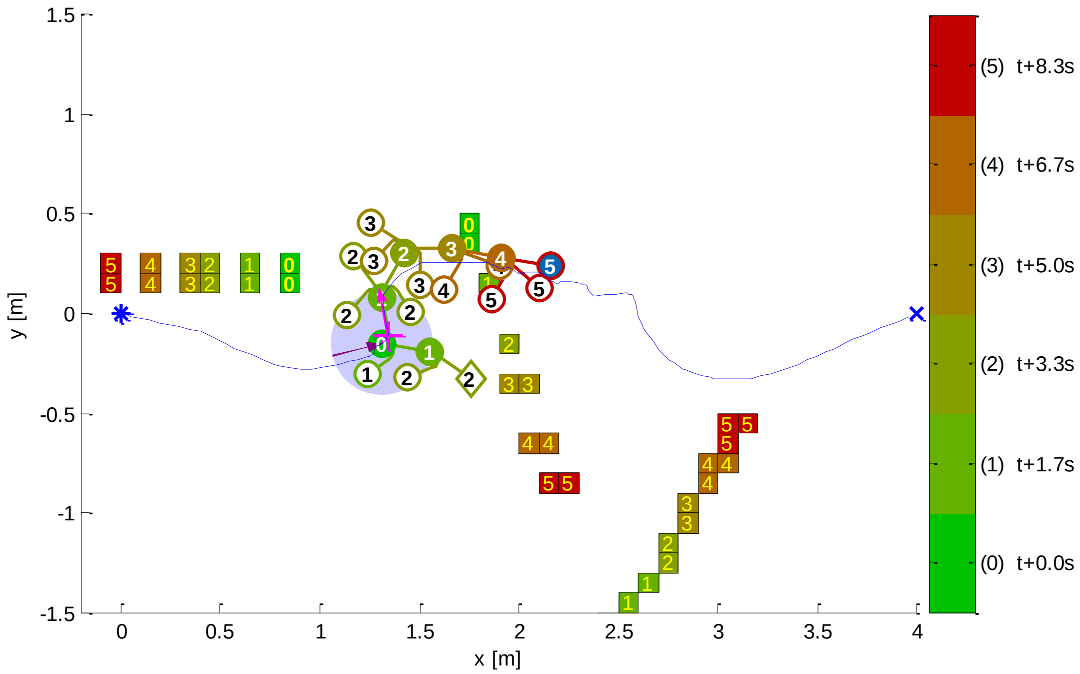
In the experiment, three moving obstacles were sent in different directions so that they should intersect expected path of the robot. In Fig. 4, there are shown three instants of time with visualized data that had been recorded during the experiment. The corresponding look-ahead tree is shown for each instant of time. The development of the tree was affected by projections of the moving obstacles. It is clear that optimal branch is bending in such way that the algorithm leads the robot to avoid the obstacles in their predicted positions.

A performance of the proposed method is compared with original VFH*. It is obvious, that VFH* in Fig. 5 failed because it led the robot into the collision with the mobile obstacle in time 11.5s. This unmodified method does not create projections of obstacles so it reacts only on green ones with zero labels which are in this case considered to be static obstacles.



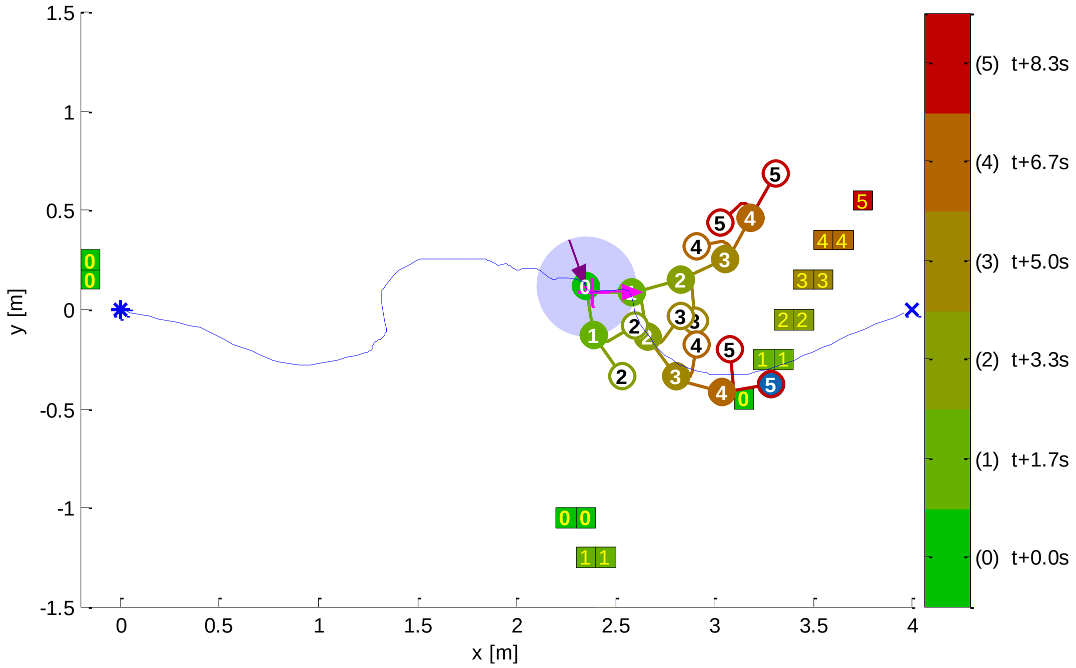
y [m]
x [m]
(5) t+8.3s
(4) t+6.7s
(3) t+5.0s
(2) t+3.3s
(1) t+1.7s
(0) t+0.0s

(a)



y [m]
x [m]
(5) t+8.3s
(4) t+6.7s
(3) t+5.0s
(2) t+3.3s
(1) t+1.7s
(0) t+0.0s

(b)



(c)

Fig. 4 Performance of improved VFH* in the presence of three moving obstacles. Three moments are displayed: a) t=0.3s , b) t=9.3s , c) t=18.7s

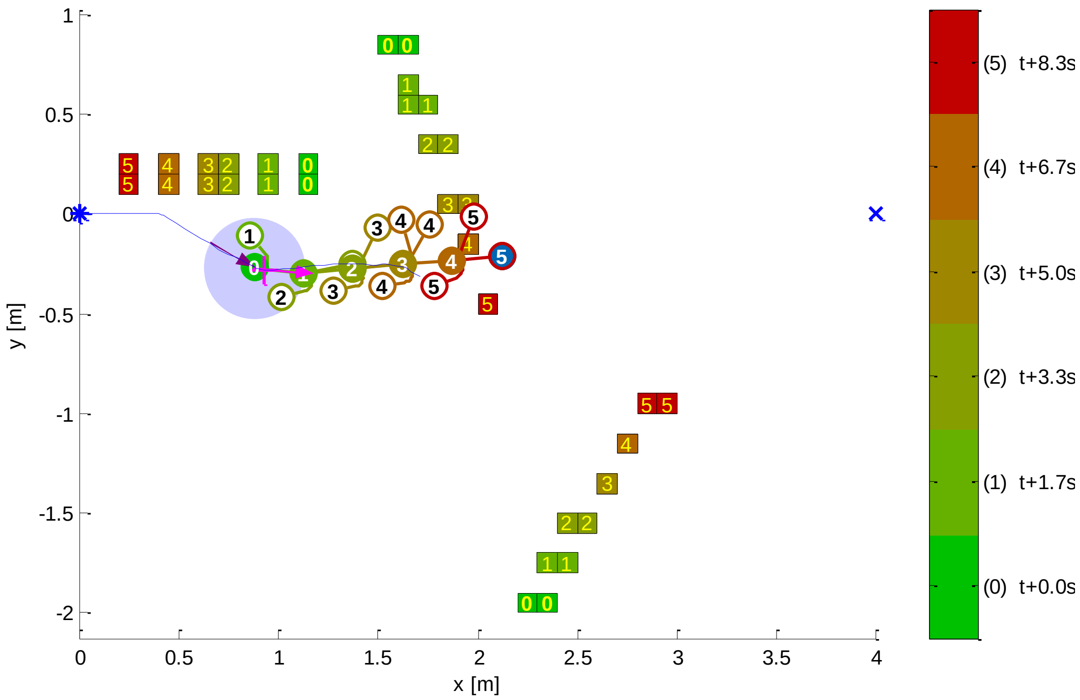


Fig. 5 Performance of original VFH* in the presence of three moving obstacles. A time stamp of the developed look-ahead tree is t=6.3s, the collision occurred in the time 11.5s.

Conclusions

The experiments in the article show that the modified VFH* is capable of collision-free navigation of the robot to the target also in environment with multiple obstacles. On the other hand, original VFH* failed because it does not take into account the time-varying environment during tree development stages. Nevertheless, mainly the navigation part of the problem is studied in this paper. Although the uniform rectilinear obstacle motion is assumed, the method is not dependent on this type of motion in general. In real world, obstacles may move along arbitrary trajectories and their movement may be predicted using some extrapolation method or using an algorithm that regards a probability of changing the obstacle direction. Therefore, further research will be focused on reliable detection of dynamic objects, their classification and prediction of their movement.

Acknowledgement

This article was created with the support of the Ministry of Education, Science, Research and Sport of the Slovak Republic by grants VEGA 1/0752/17 and VEGA 1/0065/16.

References

[1] A. Simonová, L. Hargaš, D. Koniar, Z. Loncová and M. Hrianka, Mathematical analysis of complete operation cycle of a system with two-position controller, *ELEKTRO*, Strbske Pleso, 2016, pp. 213-215, doi: 10.1109/ELEKTRO.2016.7512067

[2] Ľ. Miková and M. Kelemen, Concept of Mobile Robots for Movement in an Unknown Environment, *American Journal of Mechanical Engineering*, Vol. 4, No. 7, 2016, pp 262-265.

[3] M. Kelemen, I. Virgala, T. Kelemenová, Ľ. Miková, P. Frankovský, T. Lipták and M. Lörinc, Distance Measurement via Using of Ultrasonic Sensor, *Journal of Automation and Control*, 2015, Vol. 3, No. 3, pp. 71-74, DOI:10.12691/automation-3-3-6

[4] J. Borenstein and Y. Koren, The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots, in *IEEE Journal of Robotics and Automation*, 7/3, 1991, pp. 278-288.

[5] I. Ulrich and J. Borenstein, VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots, in *Proceedings of the IEEE International Conference of Robotics and Automation*, Leuven, Belgium, 1998.

[6] I. Ulrich and J. Borenstein, VFH*: Local Obstacle Avoidance with Look-Ahead Verification, in *Proceedings of the IEEE International Conference of Robotics and Automation*, San Francisco, CA, 2000.

[7] R. Labayrade, C. Royere, D. Gruyer and D. Aubert, Cooperative Fusion for Multi-Obstacles Detection With Use of Stereovision and Laser Scanner, in *Autonomous Robots*, Vol. 19, 2005, pp. 117-140.

[8] D. Ferguson, M. Darms, C. Urmson and S. Kolski, Detection, Prediction, and Avoidance of Dynamic Obstacles in Urban Environments, in *IEEE Intelligent Vehicles Symposium*, 2008, pp. 1149-1154.

[9] A. Ess, B. Leibe, K. Schindler and L. van Gool, Moving Obstacle Detection in Highly Dynamic Scenes, in *IEEE International Conference on Robotics and Automation*, Kobe, 2009, pp. 4451-4458.