This is an author produced version of a paper presented at IEEE International Instrumentation and Measurement Technology Conference I2MTC, Minneapolis, May 6-9 2013. This paper has been peer-reviewed but does not include the final publisher proof-corrections or proceedings pagination.

© 2013 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

# Image-Based Floor Segmentation in Visual Inertial Navigation

Guillem Casas Barceló, Ghazaleh Panahandeh, and Magnus Jansson KTH Royal Institute of Technology ACCESS Linnaeus Center Email: {gcb,ghpa,janssonm}@kth.se

Abstract—This paper presents a floor segmentation algorithm for indoor sequences that works with single grey-scale images. The portion of the floor closest to the camera is segmented by judiciously joining a set of horizontal and vertical lines, previously detected. Since the proposed method is not based on computing the vanishing point, the system can deal with any kind of indoor scenes and adapts quickly to camera movements. A second contribution is the detection of moving features for points within the segmented floor area. Based on the estimated camera ego-motion, the ground plane homography is derived. Then, the expected optical flow for the ground points is calculated and used for rejecting features that belong to moving obstacles. A key point of the designed method is that no restrictions on the camera motion are imposed for the homography derivation.

### I. Introduction

Segmenting the floor is a challenging problems in image processing and has a wide range of applications in the engineering field [1], [2]. Specular reflections and textured floors are the main difficulties faced by floor segmentation algorithms [3]. Besides, accurate approaches must deal with changes in the illumination and structure of the scene [4].

In mobile robot navigation systems, visual-based techniques are replacing historical schemes due to their simplicity and low cost. Such systems need to recognize the structure of the scene and avoid both static and moving obstacles. Therefore, floor segmentation, moving objects detection, and obstacle removal become significant tasks for guiding the robot within an environment [5], [6].

Floor segmentation is also becoming interesting to be used in ground plane-based camera ego-motion estimation in vision-aided inertial navigation systems (INS), such as [7] and [8]. Although the motion estimation in these methods is based on the ground plane features, they do not specifically address the problem of floor detection. The designed method has been conceived to be integrated into the INS presented in [9].

### A. Related work

During the past years, a large amount of work related to obstacle avoidance and ground plane detection has been done (e.g., [10], [11]). Regarding single camera techniques, Wang et al. [12] presented a region-based method that provides a local obstacle map at high resolution in real-time. In [13], a homography-based technique is introduced to estimate the ground plane normal. Then, the floor is detected by computing

plane normals from motion fields in image sequences. However, these methods are restricted to static environments. On the other hand, there are also many methods in the literature focused on outdoor environments and detection of moving vehicles [14], [15]. In [16], the background motion is estimated and compensated using an affine transform, while [17] takes advantage of knowing the camera ego-motion and exploits spatial constraints to detect the motion in a second step. Odometric information is used by Braillon et al. [18] to model the expected motion of the points on the ground plane. The location of the moving obstacles is determined by the points that do not follow this model.

However, to the best of our knowledge, only few methods specifically face the problem of floor segmentation (e.g., [2], [4]). The most similar approach to the one presented in this paper was implemented by Li and Birchfield [3]. They designed a technique, applied to single color images, that combines three visual cues for evaluating the likelihood of horizontal intensity edges to be part of the wall-floor boundary. Since their algorithm computes the vanishing point, it is restricted to typical corridor scenes and adapts slowly to camera movements.

## B. Approach overview

Unlike other single image methods, such as [3], our proposed system works with grey-scale images and does not require to compute the vanishing point. Consequently, it adapts faster to changes in camera motion and is able to deal with all types of indoor scenes. A floor polyline is defined, containing the wall-floor and floor-obstacles boundaries. In order to draw this polyline, an acute way of joining the most important lines from an edge detector is applied. Finally, a mask describing the floor area is generated. A relevant attribute of our new approach is that only the boundaries below the half image are considered, assuming that a sufficient part of the floor with enough feature points is within this region.

After segmenting the floor, feature extraction and matching is performed between consecutive frames. Based on the estimated ego-motion of the camera, the homography matrix for the ground plane is calculated. In contrast to [18], which is restricted to forward motion, the homography in our method is derived for general motion and rotation of the camera. Then, the expected optical flow is computed for features within the floor mask at the current frame. Moving features are detected

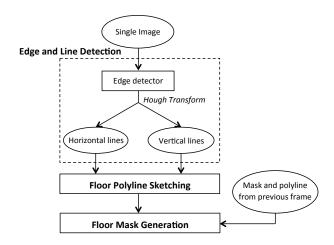


Fig. 1. General scheme for the proposed floor segmentation method.

by comparing their expected and real motion, given by the estimated optical flow and the correspondences, respectively.

### II. FLOOR SEGMENTATION ALGORITHM

As shown in Fig. 1, the proposed method is divided into three main blocks: 1) edge and line detection, 2) floor polyline sketching and 3) floor mask generation. In this section, the three parts are discussed.

## A. Edge and Line Detection

The first block of our algorithm consists of detecting, identifying and describing the main lines of the image, which define the structure of the scene. The detected lines are divided into two groups: horizontal and vertical.

First of all, the Canny edge detector [19] is applied to detect the edges from the given picture (see Fig. 2b). Then, a list of all the points belonging to every edge is generated from the Canny mask. Due to real world noisy conditions, short spurious edges that are not important regarding the scene structure description might appear. Thus, edges shorter than 60 pixels are removed at this point. Since the algorithm to draw the floor polyline requires straight vertical and horizontal lines, each edge is fitted into a set of straight segments. In order to prune and classify the line segments, the Hough transform is applied [20]. In the transformed domain, line segments are divided into two sets: vertical (listV) and horizontal (listH). Based on the tested sequences, a slope range to classify the line segments is determined. A line is classified as vertical if its slope is within  $\pm 10^{\circ}$  of the vertical direction. Horizontal segments are given a wider slope range:  $\pm 65^{\circ}$  of the horizontal direction. The rest of the lines are rejected.

In ground plane-based ego-motion estimation approaches, the ground features closest to the camera have the main contribution to the motion estimation. Since our method is designed for such applications, it is not supposed to segment the whole floor, but only a sufficient part that is the closest to the camera. Hence, lines above the half of the image will not be taken into account, reducing the computational cost of the algorithm. Applying the half image constraint entails:

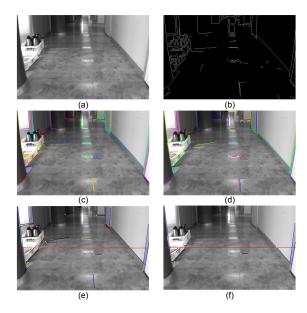


Fig. 2. Partial images from Floor Segmentation Algorithm. (a) Original image. (b) Canny edge detector mask. (c) Detected edges, each edge is painted in a random different color. (d) Line segments from detected edges. (e) Horizontal and vertical lines after classifying them in Hough domain, red line is at  $\frac{1}{2}$  of the image. (f) Remaining lines after applying the half image constraint.

- Vertical lines whose bottom points lie above it are removed.
- Vertical lines with both top and bottom points below it are removed.
- Horizontal lines whose beginning and ending points lie above it are removed.
- Horizontal lines with just one beginning/ending point below it are cut. The new beginning/ending point is set at the point of the line that corresponds to the y coordinate value equal to one half of the height of the image.

# B. Floor Polyline Sketching

A polyline representing the wall-floor and floor-obstacles boundaries at the bottom half image is drawn by judiciously joining lines from the two lists defined in Section II-A. Knowledge of the height and orientation of the camera, as well as typical structure of indoor scenes and geometric constraints are taken into account to select which lines to use. The main idea is to draw a polyline, from left to right, connecting the endings of the horizontal and vertical lines one progressively encounters.

For every iteration in the main algorithm (see Fig. 3), the first step is to find which point, within listH and listV, is most to the left. While there is still some element in any list, the line that has its beginning point (in the case of horizontal lines) or its bottom point (in the case of vertical lines) most to the left is selected. During all the procedure, the coordinates of the last point of the line segment that is being drawn are stored (lastX, lastY).

When a vertical line is chosen, its bottom point is directly used for drawing the floor polyline. However, in order to

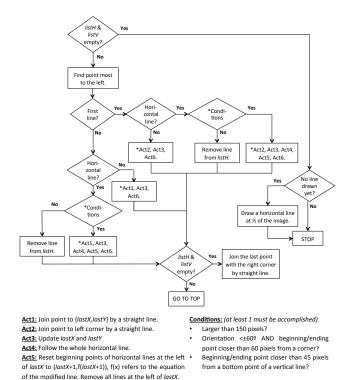


Fig. 3. Flowchart of the Floor Polyline Sketching algorithm.

avoid the effect of spurious edges that might appear because of textured floors or specular reflections, three conditions are checked when a horizontal line is selected. At least one of these conditions must be satisfied to consider the horizontal line as a segment of the floor polyline. The two first ones are related to how the wall-floor boundary should look like, while the third one is to ensure obstacle detection. The first condition arises because horizontal lines that are part of the wall-floor boundary are expected to be long. Due to the height and the orientation of the camera, they are also expected to describe a certain angle with respect to the vertical direction, which fulfills the requirements of the second condition. For possible obstacles, both the horizontal line of its base and the vertical lines of its edges are detected, so the third condition holds.

When all the lines have been analyzed, the last considered point is joined with the right corner of the image by a horizontal straight line, giving the final floor polyline. If no line has been used so far, the whole bottom part of the image is assumed to be part of the ground. Consequently, the floor polyline becomes a straight horizontal line at one half of the height of the image.

# C. Floor Mask Generation

Act6: Remove line from list.

A first version of the mask is generated by setting the pixels below the polyline to white. The rest of the pixels in the image remain in black. The area of the floor region is computed by summing-up the number of white pixels within the mask  $Npix_t$ . In order to avoid sudden changes that might reduce dramatically the floor area,  $Npix_t$  is compared with

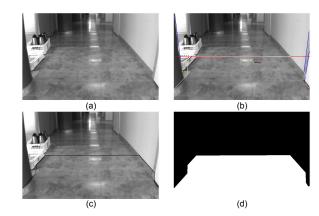


Fig. 4. Floor polyline and mask generation. (a) Original image. (b) Lines used to define the floor polyline. (c) Final floor polyline. (d) Final floor mask.

the area defined by the mask at the previous frame  $Npix_{t-1}$ . If the new area is more than 30% smaller than the area of the floor region at the previous frame  $Npix_t < 0.7Npix_{t-1}$ , the method keeps the same floor polyline and the same mask as the previous frame. Real obstacles do not appear fast enough in order to reduce more than 30% of the region of the floor from one frame to the next one. On the contrary, changes in the illumination of the scene or textured floors can cause this effect and the system must ignore them.

### III. MOVING FEATURES DETECTION

Although moving objects with well-defined edges are rejected by the floor mask, irregular moving obstacles, such as feet, might partially lie within the mask. To avoid selecting features belonging to moving obstacles, a moving features detection method for features inside the mask has been designed.

After establishing feature correspondences between consecutive frames, the expected optical flow of the ground points is compared with the real motion of each feature correspondence in order to detect moving features. Using the provided camera ego-motion estimation from [9], the homography matrix H for the ground plane is derived following a similar procedure as the one presented in [18]. Then, the expected optical flow  $\vec{f}(u,v)$  is computed from H, only for the feature correspondences inside the floor mask at the current frame. The proposed solution in [18] is derived for the specific case where the orientation of the camera relative to the ground is fixed and the motion is straight forward. In contrast to their method, here the derivation of H is generalized. The details of the this derivation are given in the Appendix.

# A. Feature Pruning

In the first step, features are removed if their optical flow vector and their correspondence vector angles differ more than a certain threshold. For the tested pairs of frames, this threshold is heuristically set to  $20^{\circ}$ . Mismatches and features with a significant different motion direction from the motion of the camera are removed right at this point. In order to prune remaining moving features, a similarity value is calculated.

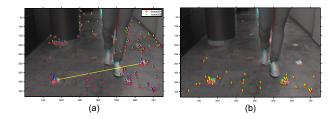


Fig. 5. Pruning of moving features. (a) All features correspondences between frames with their corresponding estimated optical flow. (b) Correspondences after applying the floor mask and pruning moving features.

To begin with, the intensity difference between every feature in the current frame and its expected position in the previous image is computed. This difference value is defined as the sum of squares differences (SSD) within a neighborhood region around the two positions:

$$D = \sum_{(i,j)\in W} [I_t(u+i,v+j) - I_{t-\Delta t}(\bar{u}+i,\bar{v}+j)]^2 \quad (1)$$

where  $I_t(u,v)$  is the intensity value of a feature at  $(u,v)^T$  in the current frame and  $I_{t-\Delta t}(\bar{u},\bar{v})$  is the intensity value of the feature in the previous image  $(\bar{u},\bar{v})^T=(u,v)^T-\Delta t\vec{f}(u,v)^T$ .

For the tested pairs of frames, the neighborhood window size is set to  $9\times 9$ , which reaches a good compromise between performance and computational time. Assuming constant illumination, D tends to zero for features on the ground, while it has greater values for features belonging to moving obstacles. Then, a similarity value for every match is defined as  $1-\bar{D}$ , where  $\bar{D}$  is the normalized difference. This implies that the similarity is between 0 and 1, giving an idea of the likelihood of each feature correspondence to be part of the ground. Finally, the similarity value is thresholded and all features whose similarity value is below a threshold are removed. This parameter regulates the amount of features that are rejected and can be optimized at every frame. For the tested sequences, it is set to 0.7, which shows a good performance (see Fig. 5).

### IV. EXPERIMENTAL RESULTS

The implemented algorithms have been tested using several forward-looking sequences reflecting typical indoor scenes challenges, such as specular reflections or textured floors. An AVT Guppy monochrome camera was used, which generated images with a resolution of 752x480 pixels, 8 bits and 10Hz. It was rigidly mounted at the top of a trolley, at 85cm height, and shifted 25° towards the floor to maximize the floor area below the half of the image. Examples of the recorded sequences as well as the floor segmentation output videos for all the tested sequences are available at the YouTube channel [21].

In order to quantitatively evaluate the performance of the floor segmentation algorithm, the following test was carried out. Up to 500 random frames from the set of recorded sequences were manually labeled yielding a ground truth wall-floor and floor-obstacles boundary at the bottom half part of the image. The error of the algorithm is defined as

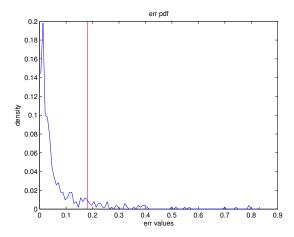


Fig. 6. Error measurement (err) pdf for the 500 labeled images. The red vertical line is situated at 0.18

the percentage of the area of the difference, between the ground truth and the estimated polyline, with respect to the total area of the ground truth floor. This difference can be computed as the sum, over all the columns in the image, of the difference between the y coordinates of the ground truth and the estimated polyline as:

$$err = \frac{\sum_{x} |\hat{y}(x) - y_{GT}(x)|}{Npix_{GT}}$$
 (2)

Fig. 6 depicts the probability density function of err over all the labeled frames. Notice that, for most of the tested frames, the err has a very low value. Considering the floor polyline estimation as a success if err < 0.18, our approach correctly detects the floor region in the 89.8% of cases.

Fig. 7 presents a set of sample output frames of the algorithm. These results prove that our approach can be applied in different environments, even with the presence of specular reflections, changes in the illumination and textured floors. Besides, it is able to deal with big moving obstacles with well-defined edges. However, tests have proved that the floor segmentation method itself is not able to reject small irregular obstacles, such as people's feet, from the floor mask (see Fig. 8). Hence, a moving features detection method has been implemented as well.

On the contrary, no quantitative performance evaluation has been carried out for the moving features detection. Tests have proved that the optical flow estimation fails in around 40% of the analyzed frames. Nevertheless, when a correct estimation is achieved, the moving features pruning method, together with the floor mask, is able to remove more than 90% of the features belonging to moving obstacles (see Fig. 5).

### V. CONCLUSIONS AND FUTURE WORK

This paper has presented a method to segment the floor and detect moving features. The floor segmentation algorithm can deal with difficult illumination conditions, specular reflections and static obstacles. Moreover, it has been proved to be able to adapt quickly to changes in the scene. The estimated motion parameters of the camera are assumed to be known for detecting moving features. A noteworthy contribution of the designed method is that no restrictions on the camera motion are introduced for the ground plane homography estimation. However, results have proved that the optical flow estimation only holds in around 60% of the tested examples. So far, no pattern is detected in the erroneous estimations. A deep analysis must be carried out to find the causes of this low percentage of effectiveness. We believe that exploiting temporal information in the homography estimation would increase the performance.

### APPENDIX

# HOMOGRAPHY AND OPTICAL FLOW DERIVATION

The projection equation for general motion of the camera of 3D points into the image pixel coordinates is:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix}_{Image} = KR_b^c \left[ R_n^b - R_n^b \vec{p} \right] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}_{World} \tag{A.3}$$

where K is the matrix of the intrinsic camera parameters;  $R_h^c$ is the direction-cosine matrix that rotates a vector from camera to body-pixels and it is constant all along the sequence; vector  $\vec{p}$  contains the estimated position of the camera at the current frame;  $R_n^b$  is the matrix that rotates from body to navigationframes. It is defined by the angles of the rotated coordinate system (equation (2.31) in [22]) as:

$$R_n^b = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix}$$
(A.4)

$$R_{11} = \cos(\psi)\cos(\theta) \tag{A.5a}$$

$$R_{12} = \sin(\psi)\cos(\theta) \tag{A.5b}$$

$$R_{13} = -\sin(\theta) \tag{A.5c}$$

$$R_{21} = -\sin(\psi)\cos(\phi) + \cos(\psi)\sin(\theta)\sin(\phi) \qquad (A.5d)$$

$$R_{22} = \cos(\psi)\cos(\phi) + \sin(\psi)\sin(\theta)\sin(\phi)$$

$$(A.5e)$$

$$R_{23} = \cos(\theta)\sin(\phi) \tag{A.5f}$$

$$R_{31} = \sin(\psi)\sin(\phi) + \cos(\psi)\sin(\theta)\cos(\phi)$$

$$R_{32} = -\cos(\psi)\sin(\phi) + \sin(\psi)\sin(\theta)\cos(\phi)$$
(A.5g)
(A.5h)

$$R_{33} = \cos(\theta)\cos(\phi) \tag{A.5i}$$

Assuming that the ground is flat and located at Z = 0:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix}_{Image} = H \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}_{Ground}$$
 (A.6)

where H is defined by:

$$H = KR_b^c J \tag{A.7}$$

and

$$J = \begin{pmatrix} R_{11} & R_{12} & -R_{11}p_x - R_{12}p_y - R_{13}p_z \\ R_{21} & R_{22} & -R_{21}p_x - R_{22}p_y - R_{23}p_z \\ R_{31} & R_{32} & -R_{31}p_x - R_{32}p_y - R_{33}p_z \end{pmatrix}$$
(A.8)

Now the temporal derivative of equation (A.6) is computed:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix}_{Image} = \dot{H} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}_{Ground}$$
 (A.9)

The final relation, combining equations (A.6) and (A.9), becomes:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix}_{Image} = \dot{H}H^{-1} \begin{pmatrix} u \\ v \\ w \end{pmatrix}_{Image}$$
 (A.10)

where

(A.5e)

$$\dot{H} = KR_b^c \dot{J} \tag{A.11}$$

By definition, the optical flow can be expressed as:

By definition, the optical flow can be expressed as: 
$$\vec{f}(u,v,w) = \left( \begin{array}{c} (\frac{\dot{u}}{w}), & (\frac{\dot{v}}{w}) \end{array} \right) = \left( \begin{array}{c} \frac{\dot{u}w - u\dot{w}}{w^2}, & \frac{\dot{v}w - v\dot{w}}{w^2} \end{array} \right) \tag{A.12}$$

The following transformations are used:

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \frac{1}{w} \begin{pmatrix} u \\ v \end{pmatrix} \text{ and } \begin{pmatrix} \dot{u}' \\ \dot{v}' \\ \dot{w}' \end{pmatrix} = \frac{1}{w} \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} \quad (A.13)$$

$$\begin{pmatrix} \dot{u}' \\ \dot{v}' \\ \dot{w}' \end{pmatrix} = \dot{H}H^{-1} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} \tag{A.14}$$

The final expression for the optical flow is then defined as:

$$\vec{f}(u',v') = \begin{pmatrix} \dot{u}' - u'\dot{w}' \\ \dot{v}' - v'\dot{w}' \end{pmatrix}$$
 (A.15)



Fig. 7. Floor Segmentation Algorithm performance. Random frames from three different sequences proving that the method can deal with specular reflections, textured floors and both static and moving obstacles with well defined edges. Full sequences can be found in [21].



Fig. 8. Floor Segmentation Algorithm performance. Notice that the feet are partially inside the defined floor area, in most of the cases.

### REFERENCES

- Z. Chen and S. Birchfield, "Visual detection of lintel-occluded doors from a single image," in *IEEE Computer Vision and Pattern Recognition* Workshops (CVPR), 2008, pp. 1 –8.
- [2] D. Lee, M. Hebert, and T. Kanade, "Geometric reasoning for single image structure recovery," in *Computer Vision and Pattern Recognition* (CVPR), 2009, pp. 2136 –2143.
- [3] Y. Li and S. Birchfield, "Image-based segmentation of indoor corridor floors for a mobile robot," in *Intelligent Robots and Systems (IROS)*, 2010, pp. 837 –843.
- [4] N. Pears and B. Liang, "Ground plane segmentation for mobile robot visual navigation," in *IEEE Intelligent Robots and Systems*, 2001, vol. 3, pp. 1513 –1518.
- [5] D. Conrad and G. DeSouza, "Homography-based ground plane detection for mobile robot navigation using a modified EM algorithm," in *Proc.* of ICAR, May. 2010, pp. 910–915.
- [6] G. Panahandeh, N. Mohammadiha, and M. Jansson, "Ground plane feature detection in mobile vision-aided inertial navigation," in *Intelligent Robots and Systems (IROS)*, 2012.
- [7] X. Song, L. Seneviratne, and K. Althoefer, "A kalman filter-integrated optical flow method for velocity sensing of mobile robots," *Mechatronics, IEEE/ASME Transactions on*, vol. 16, no. 3, pp. 551 –563, 2011.
- [8] C. Hide, T. Botterill, and M. Andreotti, "Low cost vision-aided imu for pedestrian navigation," in *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*, 2010, pp. 1 –7.
- [9] G. Panahandeh, D. Zachariah, and M. Jansson, "Exploiting ground plane constraints for visual-inertial navigation," in *Position Location and Navigation Symposium (PLANS)*, 2012, pp. 527 –534.
- [10] T. Wekel, O. Kroll-Peters, and S. Albayrak, "Vision based obstacle detection for wheeled robots," in *IEEE Control, Automation and Systems* (ICCAS), 2008, pp. 1587 –1592.

- [11] P. Lombardi, M. Zanin, and S. Messelodi, "Unified stereovision for ground, road, and obstacle detection," in *IEEE Intelligent Vehicles Symposium*, 2005, pp. 783 – 788.
- [12] H. Wang, K. Yuan, W. Zou, and Y. Peng, "Real-time obstacle detection with a single camera," in *Industrial Technology*, 2005, pp. 92 – 96.
- [13] Y. Kim and H. Kim, "Layered ground floor detection for vision-based mobile robot navigation," in *IEEE Robotics and Automation*, 2004, vol. 1, pp. 13 – 18.
- [14] B. Jung and G. S. Sukhatme, "Detecting moving objects using a single camera on a mobile robot in an outdoor environment," in in International Conference on Intelligent Autonomous Systems, 2004, pp. 980–987.
- [15] J. Odobez and P. Bouthemy, "Detection of multiple moving objects using multiscale mrf with camera motion compensation," in *Image Processing* (IEEE), 1994, vol. 2, pp. 257 –261.
- [16] A. Behrad, A. Shahrokni, S. A. Motamedi, and K. Madani, "A Robust Vision-based Moving Target Detection and Tracking System," in *Image and Vision Computing conference*, 2001.
- [17] J. Klappstein, F. Stein, and U. Franke, "Monocular motion detection using spatial constraints in a unified manner," in *IEEE Intelligent Vehicles Symposium*, 2006, pp. 261 –267.
- [18] C. Braillon, C. Pradalier, J. Crowley, and C. Laugier, "Real-time moving obstacle detection using optical flow models," in *IEEE Intelligent* Vehicles Symposium, 2006, pp. 466 –471.
- [19] J. Canny, "A computational approach to edge detection," Pattern Analysis and Machine Intelligence, 1986, vol. PAMI-8, no. 6, pp. 679 –698.
- [20] P. Hough, "Method and means for recognizing complex patterns," U.S. Patent 3.069.654, 1962.
- [21] Youtube channel with tested sequences. Available online from: http://www.youtube.com/channel/UC3NXaWEQBIZXsJAtNvBXSeg/videos?flow=grid&view=1.
- [22] M. B. Jay Farrell, The Global Positioning System and Inertial Navigation, 1st ed. McGraw-Hill Professional, 1998.