Observing Dynamic Urban Environment through Stereo-Vision Based Dynamic Occupancy Grid Mapping

You Li and Yassine Ruichek

Institut de Recherche sur les Transports, l'Energie et la Société, le laboratoire Systèmes et Transport (IRTES-SET), Université de Technology of Belfort-Montbéliard, Belfort, 90010, France
{you.li,yassine.ruichek}@utbm.fr

Abstract. Occupancy grid maps are popular tools of representing surrounding environments for mobile robots/ intelligent vehicles. When moving in dynamic real world, traditional occupancy grid mapping is required not only to be able to detect occupied areas, but also to be able to understand the dynamic circumstance. The paper addresses this issue by presenting a stereo-vision based framework to create dynamic occupancy grid map, for the purpose of intelligent vehicle. In the proposed framework, a sparse feature points matching and a dense stereo matching are performed in parallel for each stereo image pair. The former process is used to analyze motions of the vehicle itself and also surrounding moving objects. The latter process calculates dense disparity image, as well as U-V disparity maps applied for pixel-wise moving objects segmentation and dynamic occupancy grid mapping. Principal advantage of the proposed framework is the ability of mapping occupied areas and moving objects at the same time. Meanwhile, compared with some existing methods, the stereo-vision based occupancy grid mapping algorithm is improved. The proposed method is verified in real datasets acquired by our platform SeT-Car.

Keywords: Occupancy grid map, Moving objects egmentation, U-V disparity.

1 Introduction

In the field of intelligent vehicle, many tasks, such as localization, collision avoidance, path planning, are usually performed based on well represented maps. Occupancy grid map [18] is one of the most popular environmental representation tool that it maps the environment as a field of uniformly distributed binary/ternary variables indicating status of cells (occupied, free or undetected). In literature, range sensors, such as lidar and radar are usually used for creating occupancy grid maps. In contrast to pervasive applications of visual systems in intelligent vehicles, occupancy grid mapping by visual systems are not well researched. Several existing approaches are listed as follows. In [11], the authors regard stereo sensor as a range finder, taking the first encountered object in each column as an occupied grid. [3] clusters the detected points above the ground plane as occupied grids. Three different types of occupancy grids are analysed and compared in [2]. Quite similar to [2], the method proposed in [12] introduces an inverse sensor model for stereo-camera. In [4], occupancy grid map is generated from

A. Petrosino (Ed.): ICIAP 2013, Part II, LNCS 8157, pp. 379–388, 2013.

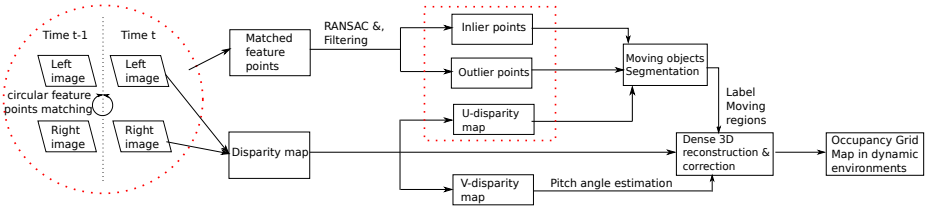


Fig. 1. Flow diagram of the proposed method

a digital elevation map after filtering out the road and isle cells. [13] directly calculates occupancy grids by several probabilistic occupancy models in obstacle U-disparity image.

However, the aforementioned papers do not treat the problem of moving objects in dynamic environments. Many researches have been conducted in this area based on range sensors (lidar, radar), such as [19]. As in computer vision, moving object detection and tracking from moving vision systems is a classic but still open research area. Proposed approaches could be roughly divided into two categories. The first uses global motion compensation to generate a background model as utilized in motion detection in static camera cases [14]. This method suffers from severe limitations in the assumption of homography transform between consecutive images. Although several improvements have been introduced in [8], [16], it is still unable to deal with complex environments. The second category of approaches are based on analyzing displacements of pixels in image plane (optical flow) [5], [15], or in 3D real world (scene flow) [10], [21]. This kind of methods usually involve joint estimation of ego-motion as well as moving objects. Advantages of the second category are no assumptions for specific environments and the ability of estimating motions at the same time.

This paper proposes a stereo-vision based occupancy grid mapping framework in dynamic environments. The proposed framework consists of two parallelly performed while mutually influenced processes: circular feature points matching between 2 consecutive stereo image pairs, and dense stereo processing. The former process is aimed to estimate motion information (self-motion and others), which conducts to a moving object segmentation in disparity map. The latter process, together with the results of motion segmentation, provide information for the final occupancy grid mapping in dynamic environments. Fig. 1 visualizes the whole framework. The contribution of this paper lies in that, other than the well known occupancy grid mapping techniques based on range sensors, it provides a framework of occupancy grid mapping based on stereo-vision in dynamic environments.

This paper is organized as follows. In Sec. 2, we introduce the sensor model of our stereoscopic system and the processing of stereo information. In Sec. 3, a moving objects segmentation method in disparity domain is described. Sec. 4 presents a probabilistic occupancy grid mapping technique, which demonstrates both obstacles and moving objects. Sec. 5 shows experimental results with real datasets. Sec. 6 summaries this paper and prospects future improvements.

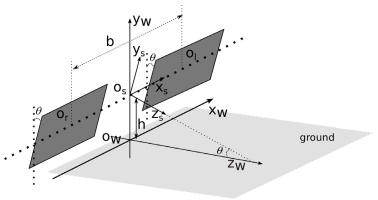


Fig. 2. Geometric model of the stereoscopic system

2 Sensor Model and Dense Stereo Vision Processing

2.1 Sensor Model

In our system, a binocular stereo-vision sensor (Bumblebee XB3) is mounted on top of a vehicle. The stereoscopic system is previously calibrated and rectified by [1]. Therefore, the left and right cameras are viewed as the same and modeled by classic pinhole model (f, c_u, c_v), where f is the focal length, (c_u, c_v) is the position of principal point. As shown in Fig. 2, ground is assumed to be a flat plane with pitch angle θ to the left and right image plane. The stereoscopic coordinate system is assumed to be originated from the middle point of baseline O_s. The world coordinate system is set as originated from the point O_w. 3D position of a point (X_s, Y_s, Z_s) in the stereoscopic coordinate system can be triangulated from its projections (u_l, v_l) and (u_r, v_r) in left and right image planes as:

$$X_s = \frac{(u_l - c_u) \cdot b}{\Delta} - b/2, \quad Y_s = \frac{(v_l - c_v) \cdot b}{\Delta}, \quad Z_s = \frac{b \cdot f}{\Delta} \tag{1}$$

where Δ is the disparity. Its corresponding coordinate value in the world coordinate system is:

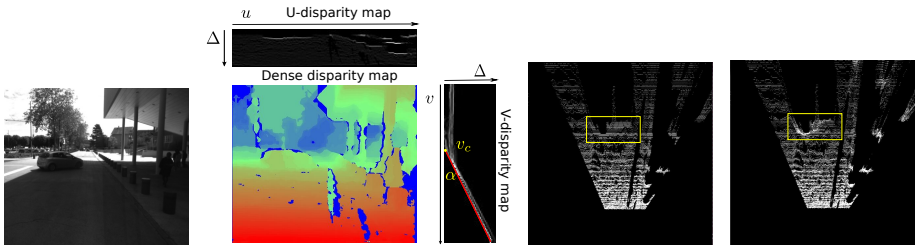
$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} + \begin{bmatrix} 0 \\ h \\ 0 \end{bmatrix} \tag{2}$$

where h is the height of camera coordinate system to the ground plane.

2.2 Dense Stereo Vision Processing

Disparity Maps. We choose Semi-Global Block Matching (SGBM) algorithm [7] to compute dense disparity image I_Δ. Fig. 3 (b) shows a disparity image calculated from Fig. 3 (a) by SGBM algorithm. Furthermore, U-V disparity maps ([20] [9] [17]) are also calculated for scene understanding. The U-V disparity images are accumulative projections of dense disparity image to the rows/columns. For example, U-disparity image is built by accumulating the pixels with the same disparity in I_Δ in each column (u-axis). V-disparity image is calculated symmetrically. Examples are shown in Fig. 3 (b). Actually, the U-disparity image could be viewed as a birds view disparity image of the scene, while in the V-disparity image, ground plane is mapped to a quasi-line

(marked in a red line in Fig. 3 (b)). Afterwards, the U-disparity image will be used for moving object detection and segmentation, while the V-disparity image will be used for estimating ground pitch angle.



(a) An image of road scene (b) Dense disparity map and U-V disparity maps (c) Projection before pitch angle correction (d) Projection after pitch angle correction

Fig. 3. Process stereo measurements

Correcting 3D Estimation. Most of the existing stereo-based occupancy grid mapping methods [2], [12] and [13] assume that the stereoscopic system is parallel to the ground. In our work, we try to correct the stereo measurements by estimating the pitch angle between the stereoscopic system and ground plane from V-disparity map. We will show that this correction improves the quality of occupancy grid map.

One attribute of V-disparity is that planar ground plane is projected into a line, as the red line drawn in Fig. 3 (b). Let the equation of ground's projection in V-disparity plane as: $V = \alpha\Delta + v_c$, where α is the slope, Δ is the disparity and v_c is the value when $\Delta = 0$. It can be inferred that a plane with equation $Z = aY + d$ in the world coordinate system is projected in V-disparity as:

$$\Delta = \frac{b}{\alpha h - d}(v - c_v)(a\sin\theta + \cos\theta) + \frac{b}{\alpha h - d} f(a\cos\theta - \sin\theta) \tag{3}$$

where θ is pitch angle between the camera coordinate system and the ground plane. For a planar ground, the pitch angle can be deduced as [9]:

$$\theta = \arctan(\frac{c_v - v_c}{f}) \tag{4}$$

In our work, the ground's projection line in the V-disparity image is extracted by Hough transform. After estimating the pitch angle, all the 3D reconstruction results acquired by triangulation are corrected according to Eq. 2. An illustrative example is shown in Fig. 3 (c) (d). The 3D estimations before and after correction are projected into grids in ground plane. The more points within one grid, the higher gray value in this grid. After correction, it is clear to see that the points belong to the vehicle (in yellow box) become obvious than before, which better represents the structure of the vehicle.

3 Moving Object Segmentation in Disparity Domain

In this section, sparse image features [6] will be extracted and matched circularly and grouped according to the estimated ego-motion.

3.1 Circular Feature Points Matching and Grouping

Circular Feature Points Matching. Sparse feature points are extracted and pairwise matched through every four images, which are the left and right images acquired at two consecutive times (Details of feature point extraction and matching are in [6]). Hence, for every consecutive stereo image pairs, we have a tuple of feature points as: $\{(u_l^t, v_l^t) \leftrightarrow (u_r^t, v_r^t) \leftrightarrow (u_l^{t-1}, v_l^{t-1}) \leftrightarrow (u_r^{t-1}, v_r^{t-1})\}$, where (u_l^t, v_l^t) is a feature point in the left image captured at time t. A bucketing process is then performed to reduce the number of matched feature points and even the distribution. Next, 3D positions of the selected feature points are calculated by Eq. 1. Since the well-known big error in 3D point triangulation in long distance, we set a region of interest (ROI), and filter out matched feature points outside this ROI.

Ego-motion Estimation and Feature Points Grouping. Here, let the left image frame be the reference coordinate system. Between two stereo image pairs in succession, the motion parameters (rotation/translation parameters), the 3D positions of a feature point P^{t-1} in time $t-1$ and its image coordinates p^t at time t are related by:

$$\tilde{p}^t = \mathbf{P} \cdot \left(\left[\mathbf{R}^{t-1} | \mathbf{T}^{t-1}\right] \cdot \tilde{P}^{t-1}\right) \tag{5}$$

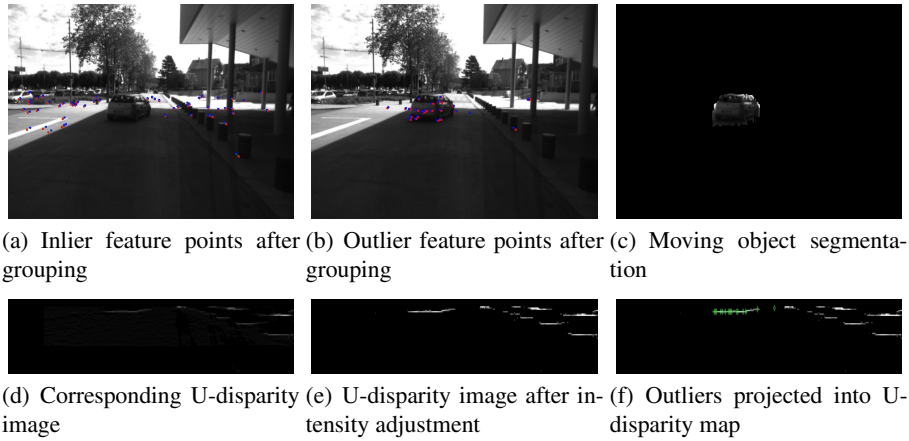
where the notation $\tilde{\cdot}$ denotes homogeneous coordinates, $\mathbf{P}$ denotes the 3×4 projection matrix of the left camera, $\mathbf{R}^{t-1}$ and $\mathbf{T}^{t-1}$ are the movement parameters of the left camera within the interval $[t-1 : t]$. Let $(P_i^t, i = 1, ..., N)$ denote 3D positions of matched feature points within ROI at time t. Egomotion parameters are estimated by Gaussian-Newton method to minimize:

$$\min \sum_{i=1}^{N} \|\mathbf{P} \cdot (\tilde{P}_i^t - [\mathbf{R}^{t-1} | \mathbf{T}^{t-1}] \cdot \tilde{P}_i^{t-1})\|^2 \tag{6}$$

Since the movements of the feature points are mainly caused by self-motion or independent objects movement, a robust statistic method, RANSAC, is applied to identify the inliers and outliers and meanwhile to estimate ego-motion parameters. The inliers are points following the movement of our experimental platform, while the outliers are points consisting of independent moving objects and noises. The feature points grouping results are shown in Fig. 4 (a),(b) (ROI is set to maximum 20 meters in distance and maximum 3 meters in height).

3.2 Moving Segmentation in U-disparity Map

According to the results of feature points grouping, it is possible to segment moving object from U-disparity map. As introduced before, U-disparity map is a projection of pixels in the dense disparity map along columns. The intensity of a pixel in U-disparity



(a) Inlier feature points after grouping (b) Outlier feature points after grouping (c) Moving object segmentation

(d) Corresponding U-disparity image (e) U-disparity image after intensity adjustment (f) Outliers projected into U-disparity map

Fig. 4. Circular feature points extraction and grouping

map $I(u, \Delta)$ represents the number of pixels with same disparity Δ in column u in dense disparity map. The most important attribute of U-disparity used for obstacle segmentation is that, an obstacle in the ground is projected as a line in U-disparity, as shown in Fig. 4 (d). However, in practice, an object close to the stereo-vision system would be always captured in more pixels than an object far from the system, this means that in U-disparity map, the objects with large disparities are always "brighter" than the objects with small disparities. Therefore, intensity adjustment for U-disparity map is necessary. We use the sigmoid logistic function to adjust the intensity of U-disparity map.

$$I' = I * \frac{r}{1 + e^{\Delta * c}} \tag{7}$$

where I' and I are the intensity of U-disparity map after and before adjustment. Δ is the disparity. r and c represent amplify factor and scale factor respectively. The advantage of sigmoid function is its ability to smoothly restrain the intensity of areas near the stereo vision system and amplify the intensity of the areas far away. After manually tuned the parameters, an example of adjusted U-disparity map is shown in Fig. 4 (e). We can see that the intensities of all potential objects are adjusted similar to each other, regardless of the distance. Based on the grouped feature points and adjusted U-disparity map, we segment the moving object from U-disparity map as follows:

1. Project all the outliers into U-disparity map based on their disparities, as shown in Fig. 4 (f).
2. Each projected outlier in U-disparity map is taken as a seed, and a flood fill segmentation approach is performed to generate a segment.
3. After obtaining all the candidate segments, a merging process is processed to merge all the segments which are mutually overlapped.
4. A post-process refinement is undertaken. For each of the candidate segments, if it contain an inlier projection in U-disparity map, it is rejected. Then, the

candidate segments that pass the refinement are assumed to be confirmed moving object segmentation results.

5. A back projection from U-disparity map to dense disparity map is performed to segment moving objects in dense disparity map. An example is shown in Fig. 4 (c).

4 Building Dynamic Occupancy Grid Map

Based on the corrected 3D points calculated by Eq. 1 and the moving objects segmentations, dynamic occupancy grid map is build within ROI. Firstly, the reconstructed 3D points are assigned to each grid according to their positions. Noticing the assumption that all the obstacles are perpendicular to the planar ground, the more number of points a grid holds, the bigger probability it is occupied. In similar, the higher average height of the points a grid holds, the more probable it is occupied. Consequently, we separately compute the occupancy probabilities $P(O|num)$ and $P(O|height)$ for each grid, where O is occupancy indicator. Then we take the weighted average of the two values as the final occupancy probability. Finally, the occupied grids, which contain 3D points from the segmented moving regions are labeled as moving areas.

Calculating Occupancy Probability from the Number of Points. When computing occupancy probability by the number of 3D points, a problem has to be dealed with is the disturbance of points on the ground. As seen in Fig. 3 (d), the grid on the ground near stereoscopic system always hold a lot of points, even more than obstacles far from stereoscopic system. In this case, direct estimating occupancy probability from number of points would be failed. In literature, [2] and [12] don't mention this problem, while [13] and [4] avoid it by a previous separation of road pixels. In our method, this problem is overcame by introducing density of the points in a grid. Let n to be the assigned number of 3D points in the grid. From the pinhole camera model, the density of reconstructed points in the grid is f/d, where f is the focal length, d is the distance to the stereoscopic system. Thus, in theory, the number of points in the grid is approximately: $n' = (f/d) \times s$, where s is the surface of the grid. In our method, a relative number of points instead of absolute number of points is used:

$$n_r = \frac{n}{n' \times \beta} \tag{8}$$

where β is a scale factor, n is the absolute detected number of points. The occupancy probability concerned about number of points is modeled as:

$$P(O|num) = 1 - e^{-(n_r/\delta_n)} \tag{9}$$

where δ_n is a scale factor. However, it is not convenient to use the probability in Eq. 9 directly. The log-odds of the probability is adopted:

$$l(O|num) = \log(\frac{P(O|num)}{1 - P(O|num)}) \tag{10}$$

Calculating Occupancy Probability from the Average Height. The average height $\bar{h}$ of 3D points in the grid could help the factor of relative number n_r when an obstacle is not perpendicular to the ground. The probability and corresponding log-odds are similar to Eq. 9 and Eq. 10.

$$P(O|height) = 1 - e^{-(\bar{h}/\delta_{\bar{h}})} \tag{11}$$

$$l(O|height) = \log(\frac{P(O|\bar{h})}{1 - P(O|\bar{h})}) \tag{12}$$

where $\delta_{\bar{h}}$ is a scale factor. The final log-odds of occupancy for a grid is a weighted average of the two estimations of Eq. 10 and Eq. 12:

$$l(O) = w_n l(O|num) + w_{\bar{h}} l(O|height) \tag{13}$$

where w_n and $w_{\bar{h}}$ are the weights and $w_n + w_{\bar{h}} = 1$. Based on the log-odds of each grid, its occupancy indicator O is decided as:

$$O = \begin{cases} undected & if\ n_r < n_t \\ occupied & if\ l(O) > l_t \\ free & if\ l(O) < l_t \end{cases} \tag{14}$$

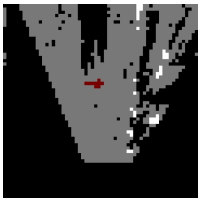
where n_t and l_t are thresholds manually set for making decision.

5 Experiments in Real Environment

The mapping method is tested in datasets acquired by our experimental vehicle in the city of Belfort, France. A stereoscopic system (Bumblebee XB3) is mounted on the top of the vehicle. The left and right cameras in Bumblebee XB3 are used with a baseline of 24cm and an image resolution of 640×480. The region of interest (ROI) for the grid map is set to $20m \times 20m$, with maximum height $3m$. The parameters used to calculate the occupancy indicator are set as: $\beta = 0.01, \delta_n = 0.2, \delta_{\bar{h}} = 0.1, w_n = w_{\bar{h}} = 0.5, n_t = 1.5, l_t = 7, r = 8, c = 0.02$.

Fig. 5 shows performances of the proposed method in 3 typical video sequences. The images in the left column are perceived by our stereoscopic system in urban environments. The middle column shows the corresponding dynamic occupancy grid mapping results. Dynamic occupied areas are labelled in red, the white and gray grids represent occupied static areas and free space respectively, while black grids are undetected areas. The tables represent error evaluation of the proposed moving objects detection/segmentation method, where "TP", "FP" are short for "true positive" (detect and segment moving objects successfully), "false positive" (taking static objects as dynamic objects, which usually caused by noisy matched feature points). From the results, we could see that the proposed method performs well in the first two sequences while declines in the third. The major reason is that a moving pedestrian is slower than a moving vehicle, and hence makes dividing dynamic/static feature points more difficult. Several feature points belong to moving pedestrian are wrongly clustered as static points. The proposed method is based on OpenCV and implemented in C++. It is tested using a laptop with a CPU Intel i3-2350 2.30GHz. Without any acceleration technique, the total computation time is 0.8 second in average.

(a) Road scene from Seq. 1

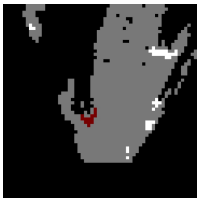


(b) Dynamic occupancy grid map

Sequence 1	TP	FP
80 frames	97.5%	10%

(c) Performance

(d) Road scene from Seq. 2

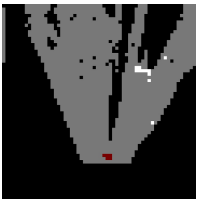


(e) Dynamic occupancy grid map

Sequence 2	TP	FP
120 frames	92.5%	8.5%

(f) Performance

(g) Road scene from Seq. 3



(h) Dynamic occupancy grid map

Sequence 3	TP	FP
140 frames	76.4%	10.7%

(i) Performance

Fig. 5. Experimental results in real urban environment

6 Conclution and Future Works

In this paper, we present a framework of dynamic occupancy grid mapping technique. The framework mainly consists of a moving object segmentation step and an occupancy probability estimation step. The moving object segmentation is conducted with circularly matched feature points, while the occupancy grid mapping is performed with dense stereo information. In future, we are planning to strengthen the robustness of the moving object segmentation, as well as to improve smoothness of occupancy grid mapping.

References

1. Camera calibration toolbox for matlab (2012), `http://www.vision.caltech.edu/bouguetj/calib_doc`
2. Badino, H., Franke, U., Mester, R.: Free space computation using stochastic occupancy grids and dynamic. In: Programming, Proc. Intl Conf. Computer Vision, Workshop Dynamical Vision (2007)

3. Braillon, C., Pradalier, C., Usher, K., Crowley, J., Laugier, C.: Occupancy grids from stereo and optical flow data. In: Proc. of the Int. Symp. on Experimental Robotics (2006)
4. Danescu, R., Oniga, F., Nedevschi, S., Meinecke, M.M.: Tracking multiple objects using particle filters and digital elevation maps. In: IEEE Intelligent Vehicles Symposium, pp. 88–93 (2009)
5. Dey, S., Reilly, V., Saleemi, I., Shah, M.: Detection of independently moving objects in non-planar scenes via multi-frame monocular epipolar constraint. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 860–873. Springer, Heidelberg (2012)
6. Geiger, A., Ziegler, J., Stiller, C.: Stereoscan: Dense 3d reconstruction in real-time. In: IEEE Intelligent Vehicles Symposium. Baden-Baden, Germany (June 2011)
7. Hirschmueller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Trans. Pattern Analysis and Machine Intelligence 30, 328–341 (2008)
8. Kang, J., Cohen, I., Yuan, C.: Detection and tracking of moving objects from a moving platform in presence of strong parallax. In: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV) (2005)
9. Labayarde, R., Aubert, D., Tarel, J.P.: Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. In: IEEE Intelligent Vehicle Symposium, vol. 2, pp. 646–651 (2002)
10. Lenz, P., Ziegler, J., Geiger, A., Roser, M.: Sparse scene flow segmentation for moving object detection in urban environments. In: IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, pp. 926–932 (2011)
11. Murray, D., Little, J.J.: Using real-time stereo vision for mobile robot navigation. Autonomous Robots 8, 161–171 (2000)
12. Nguyen, T.N., Michaelis, B.: Al-Hamadi: Stereo-camera-based urban environment perception using occupancy grid and object tracking. IEEE Transactions on Intelligent Transportation Systems 13, 154–165 (2012)
13. Perrollaz, M., John-David, Y., Anne, S., Laugier, C.: Using the disparity space to compute occupancy grids from stereo-vision. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (October 2010)
14. Kaucic, R., et al.: A unified framework for tracking through occlusions and across sensor gaps. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) (2005)
15. Rabe, C.: Fast detection of moving objects in complex scenarios. In: IEEE Intelligent Vehicles Symposium (2007)
16. Sawhney, H.: Independent motion detection in 3d scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 1191–1199 (2000)
17. Soquet, N., Aubert, D., Hautiere, N.: Road segmentation supervised by an extended v-disparity algorithm for autonomous navigation. In: IEEE Intelligent Vehicle Symposium, pp. 160–165 (June 2007)
18. Thrun, S.: Learning Occupancy Grid Maps with Forward Sensor Models. Autonomous Robots 15, 111–127 (2003)
19. Dung, V.T., Aycard, O.: Online localization and mapping with moving object tracking in dynamic outdoor environments. In: Proceedings of the IEEE Intelligent Vehicles Symposium (2007)
20. Wang, J., Hu, Z., Lu, H., Uchimura, K.: Motion detection in driving environment using U-V-disparity. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3851, pp. 307–316. Springer, Heidelberg (2006)
21. Wedel, A., Meißner, A., Rabe, C., Franke, U., Cremers, D.: Detection and segmentation of independently moving objects from dense scene flow. In: Cremers, D., Boykov, Y., Blake, A., Schmidt, F.R. (eds.) EMMCVPR 2009. LNCS, vol. 5681, pp. 14–27. Springer, Heidelberg (2009)