ACCESSIBLE SECTION DETECTION FOR VISUAL GUIDANCE

Daniel Koester, Boris Schauerte, Rainer Stiefelhagen

Karlsruhe Institute of Technology
Department of Computer Science
Karlsruhe, Germany
{daniel.koester,boris.schauerte,rainer.stiefelhagen}@kit.edu

ABSTRACT

We address the problem of determining the accessible section in front of a walking person. In our definition, the accessible section is the spatial region that is not blocked by obstacles. For this purpose, we use gradients to calculate surface normals on the depth map and subsequently determine the accessible section using these surface normals. We demonstrate the effectiveness of the proposed approach on a novel, challenging dataset. The dataset consists of urban outdoor and indoor scenes that were recorded with a handheld stereo camera.

Index Terms— Visually Impaired, Computer Vision, Obstacle Detection, Guidance, Navigation

1. INTRODUCTION

Assisting visually impaired people to safely navigate and explore urban areas is an essential task when aiming toward increasing their autonomy, mobility, and overall life quality. While location and directionality information provided by the global positioning system (GPS) can guide users toward points of interest, GPS is blind with respect to the user's immediate surroundings. Consequently, we require complementary systems to perceive the user's surroundings and inform him about potential obstacles and dangers in his path.

Many systems have been developed and can be used to detect obstacles, including the classical white cane. Most technical solutions are often targeted toward different applications, imposing specialized constraints. Furthermore, they often rely on specialized, costly hardware such as sonar, radar, or light detection and ranging (LIDAR). This hardware is incapable of perceiving information provided by, e.g., traffic lights, signs, and lane markings. In contrast, the nowadays omnipresent cameras are able to perceive such information, they are extremely cheap, and the field of computer vision has made tremendous progress in the last decade.

In this paper, we present a computer vision approach to detect the accessible section in front of the user. Compared to the classical white cane, this has the advantage that we can guide the user more smoothly around obstacles, because we perceive these at a greater distance. In contrast to many existing approaches that model and try to detect obstacle classes, we investigate the dual problem of determining the parts in an image that are not blocked by obstacles. To this end, we use depth information to determine the ground plane and then determine the accessible section under the assumption that the area directly in front of the user is accessible. This is a feasible assumption, because since the user is constantly guided around obstacles, there should be no obstacles directly in front. Although being computationally lightweight and simple, our approach provides a consistently good performance, which we demonstrate on a novel dataset. The dataset comprises of 20 videos that cover different urban scenes and contain realistic ego-motion, lighting conditions, and scene complexity.

2. RELATED WORK

Many navigational aids for visually impaired people have already been created. The *GuideCane* [1] replaces the traditional cane with a digitally enhanced counterpart. While it is similar to the regular cane, it has a two wheeled base and an array of distance sensors mounted to it. Instead of performing a sweeping motion, the user simply pushes the GuideCane along. Detected obstacles are evaded by breaking one of its wheels, thus initiating a circular motion to avoid the obstacle. Another method uses sonar sensors and haptic feedback through small vibration units sewn into the wearer's garment and provides an unobtrusive and almost invisible way to signal feedback to the user [2].

Other systems rely on the existence of specific markers or real world characteristics. Coughlan and Maguchi [3] use colored markers installed throughout a building. These are detected by a mobile phone application to help location- and way-finding inside buildings. This improves location aware systems where GPS is not available. Markers must be provided, the layout of a site must be known and moving obstacles, such as people, are not considered. Chen et al. [4] alleviate the need of such markers by including an Inertial Measurement Unit (IMU). It is used to sample the user's kinematic data and therefore its ego-motion. By combining information

about walking direction, step length and frequency, a position estimation is created on an a priori known map.

Different approaches focus on specific subsets of obstacles. Martinez and Ruiz [5] warn of aerial obstacles only, e.g. branches or low hanging street signs. Their work complements the traditional walking stick, since such obstacles are not sensed because of the way the walking stick is used. Lee et al. [6] use saliency maps and stereo vision to segment obstacles that have a high saliency. This is especially problematic for objects that have a similar color and structure as their surroundings, e.g. curbs. Also, the authors use a *Time Of Flight* camera, as well as an *RGB-D* camera in [7], which returns depth information and thus compensates for the costly depth calculation.

General ground plane and obstacle detection can be achieved through plane fitting, as has been done by Se and Brady [8]. A linear relationship between image pixel coordinates and ground plane disparity is used. Through Random Sample Consensus (RANSAC) [9] and a Sobel edge detector the ground plane as well as the camera pose are estimated. Staircase detection is performed by Hoon et al. [10]. They rely on a trained classifier and use RANSAC to estimate the ground plane and remove false detections. Segmentation is another technique to detect the ground plane and is used by Lombardi [11].

Obstacle detection usually focuses on a subset of all possible obstacles only. Labyrade [12] and Braillon [13] use varying techniques such as the Hough Transformation [14] as well as optical flow [15] to detect prominent or salient objects.

Many works deal with pedestrian detection in urban settings, some even on a wearable platform [16], where depth templates of upper bodies are learned and matched. A similar system uses stereo camera rigs mounted onto wheeled vehicles [17, 18]. This results in steady camera movements with only very few pitch and roll changes. A probabilistic approach is then used to model the dependency of person detection, size and location, which then creates a common ground plane estimation.

3. SYSTEM DESIGN

Our accessible section detection system is based on a framework aimed toward collaborative development of assistive systems for visually impaired people. We will describe the framework further in chapter 4.1. We have encapsulated the proposed accessible section detection system inside a module of our framework, to allow for further integration and reuse. An input module delivers stereo images taken from pre-recorded sessions, e.g., the dataset, or live camera input from a calibrated stereo camera. Using a stereo reconstruction library [19], we calculate a disparity map. A *Kinect* can also be used for input and allows us to achieve real time performance. Using our disparity map, we compute the accessible section by estimating surface directions of real world scene patches.

4. ACCESSIBLE SECTION DETECTION

4.1. The Blind And Visually Impaired Support System

We created a framework, the *Blind and Visually impaired Support system (BVS)*, to simplify and accelerate collaborative research efforts toward assistive systems for visually impaired people. Our framework is inspired by the Robotic Operating System (ROS) [20]. We designed it to be fast to learn and easy to use. Furthermore, we integrated advanced functionalities into the framework, such as multi-threading, module pooling and module hot swapping. The framework itself will be made open source, as well as the base modules that help with tasks like image and video acquisition and camera calibration.

4.1.1. Modularity and Extensibility

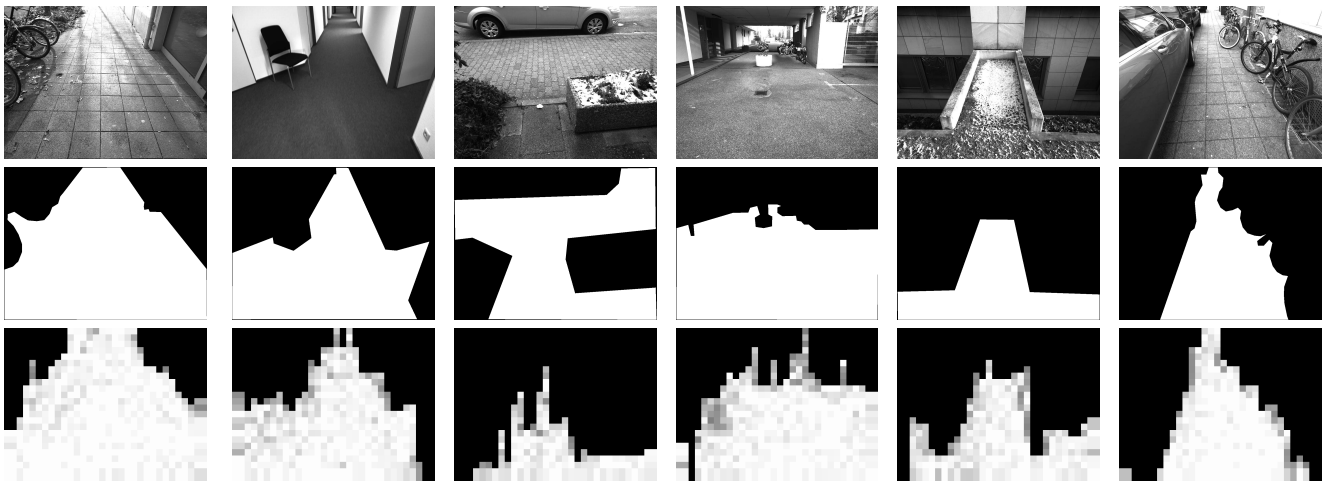
Our framework provides a modular abstraction. It allows us to separate functionalities while keeping the ability to freely combine them. To that regard, each framework module consists of a core functionality. This functionality represents its main contribution to the overall system while the module itself deals with more complicated procedures, e.g., initialization and shutdown of hardware components. We can thus combine functionalities of several modules without having to deal with their side effects. A modular approach furthermore allows us an easy extensibility of created systems. We can include missing functionalities by creating them or reusing them from other projects. Our framework approximates the running costs of a specifically created application by using a pure functional approach and a flat calling hierarchy, but keeps the benefit of being dynamically changeable in its execution details and their order, even at runtime.

4.1.2. Public Interfaces

To simplify the frameworks usage, we created various interfaces. These are aimed at the development of framework clients and especially modules. We chose to allow the development of framework clients to enable a broad usage scenario and framework portability, e.g., headless clients on servers or embedded systems, graphical user interfaces or clients for mobile operating systems. To promote a consistent module interface, we include a tool to create a module template. This template allows for easy and fast creation of library wrappers and encapsulation of functionalities. Our interfaces allow modules to connect to each other and make use of our provided configuration, information and logging subsystems.

4.1.3. Base Modules

Our framework itself contains no specific low level driver components or dependencies on other libraries. We provide such functionality through encapsulation inside modules, which we



(a) (b) (c) (d) (e) (f)

Fig. 1: Key frames, binary masks and accessible section predictions of videos (a) Sidewalk, (b) Corridor, (c) Flower-box, (d) Passage, (e) Ridge, and (f) Narrow.

release in combination with the framework. To support development toward assistive systems using computer vision, we created modules to capture, record and playback information from cameras, images and videos, calibrate stereo cameras, as well as reconstruct disparity maps. These base modules also serve an exemplary purpose for the creation of further modules.

4.2. Gradient Calculation

The local orientation of a real world surface segment correlates with the gradient of its regional representation in our disparity image. It is geometrically proven that real world points with a smaller distance toward the camera base have a greater disparity than points further away [21]. Thus a planar surface with a normal directly aimed at the projection center of the camera shows no gradient in the disparity representation, as its surface points are all approximately at the same distance. On the contrary, a tilted surface shows a gradient, as the distance of its surface points vary and therefore the calculated disparity varies. We calculate the gradient of a region in the disparity map as

$$\nabla f = \frac{\delta f}{\delta x}\hat{x} + \frac{\delta f}{\delta y}\hat{y} \quad , \tag{1}$$

which includes information about strength and direction. We then use a discrete version of the ∇ operator, a filter kernel on a small image region. The ∇ operator is separated into ∇X and ∇Y, horizontal and vertical fraction, which we calculate as

$$\nabla X = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} f(I) \, , \nabla Y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} f(I) \, , \tag{2}$$

where $f(I)$ represent a disparity map region of appropriate size.

Different kernel sizes can detect differently sized surfaces. We chose a kernel size of 32×32 pixels in our system. To decrease the computational costs, we do not apply the kernel to every pixel, but only do so block wise. We compute only a single kernel size, but further scaling of the kernel size is necessary to improve the detection, especially on the accessible section's edges, but also to detect, e.g., stair cases.

Using ∇X and ∇Y, we calculate a surface region's orientation as

$$\theta = \arctan \frac{\nabla Y}{\nabla X}. \tag{3}$$

The calculation considers only two dimensions, but is sufficient for our case, as only surface regions being completely or almost upright are of interest to us. These correspond to a gradient value θ of around $\pi/2$, or $90°$.

4.3. Accessible Section

Recording a real world scenario with a camera system imposes a few geometric constraints on the retrieved images. The accessible section is usually connected to the bottom image border, guaranteed by the fact that the cameras record a persons point of view. Furthermore, the camera is usually upright, with a varying degree of rotational deviation. This has the result, that the accessible section is also usually upright, as it mostly consists of the ground plane in front of a person. By adhering to these constraints, we can determine the accessible section.

We process the calculated surface angles from the image's bottom edge to its top edge. Starting on the image bottom, we collect all regions whose calculated angle deviates less than $\pi/8$ from an optimal upright angle of $\pi/2$. This detection step returns the *directly* accessible section, the part of the ground

plane that can be reached without having to navigate around or behind obstacles, from the current point of view.

5. EXPERIMENTAL EVALUATION

5.1. Dataset

We created a dataset to evaluate our ground detection system and will make it publicly available alongside the software framework. Existing datasets regarding any form of ground detection usually focus either on road scenes or people detection inside pedestrian areas, as discussed in section 2. Our dataset focuses on the observation that existing systems do not handle realistic ego-motion well, since they were usually not built toward a wearable platform.

Our dataset consists of 20 videos with varying length. These cover common urban scenes, e.g., walkways and sidewalks; static obstacles, e.g., parked cars, bicycles and street poles; as well as moving obstacles, e.g., cyclists and pedestrians. We also included a few rare or uncommon cases, such as a narrow ridge and a ladder like sculpture. We present some example frames of the videos contained in the dataset figure 1.

5.1.1. Acquisition

We recorded the videos using a handheld stereo camera rig and a laptop. The cameras we used were *Point Grey Grasshopper 2* cameras, which we mounted onto a metal carrier at a fixed distance and angle. We then took several precautionary measures to improve the stereo reconstruction process. First, the cameras optical axes were aligned to be parallel. Furthermore, *daisy-chaining* allows for a built-in synchronization and adaptation of gain and exposure between the cameras. We obtained the calibration of the stereo camera system before and after the recordings to mitigate unintended changes in the stereo base due to external forces and detected no significant changes. Videos were recorded at a resolution of 1024×768 pixels and 15 frames per second in an 8-bit grey mode.

Much care was taken to record the videos under realistic settings. They include strongly lit and shadowed regions, lens flares and realistic ego-motion on all axes, as they were recorded on a handheld mobile platform carried by a pedestrian.

5.1.2. Ground truth

Overall, the dataset contains 7789 frames. Of these, we labeled every fifth frame, which amounts to three labeled frames per second. To allow the cameras to properly synchronize gain and exposure, we did not include the first 30 frames of each video. Then two persons marked the desired accessible section with a polygon, labeling different videos each.

We imposed fixed constraints on the labeling process. A valid accessible section must connect to the bottom frame boundary to be reachable from the current position, otherwise obstacles could obstruct it. We show some examples of binary masks created from labeled frames in figure 1.

5.2. Measures

We evaluate the performance of the created ground detection system by calculating the *Receiver Operating Characteristic (ROC)*, the *Precision-Recall (PR)* as well as aggregated measures in form of the *Area Under the Curve (AUC)* and F_β scores. The F_β score combines precision and recall into a single value, their individual weights determined by β,

$$F_\beta = (1+\beta^2) \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{(\beta^2 \cdot \mathrm{precision}) + \mathrm{recall}} \quad . \tag{4}$$

When evenly weighted, the F-score becomes the balanced F-measure, also known as the F_1 score. Additionally, in the following we will use $\mathrm{F}_{0.5}$, as it places a higher importance on precision than recall. A high precision seems more relevant than a high recall in this application, since it correlates with a higher percentage of correct results with fewer false positives. This is especially important when dealing with a system that directly affects a human being, as it seems preferable to detect all obstacles rather than all of the accessible section.

Finally, we calculate the overall pixel-wise accuracy for each video as

$$\mathrm{accuracy} = \frac{\#\mathrm{TP} + \#\mathrm{TN}}{\#\mathrm{TP} + \#\mathrm{TN} + \#\mathrm{FP} + \#\mathrm{FN}} \quad , \tag{5}$$

with true-positives (TP), true-negatives (TN), false-positives (FP) and false-negatives (FN).

5.3. Results

Our created accessible section detection achieves varying results depending on circumstances like the amount of observable ground plane or its texture. In these and similar scenarios, where there is a great variability in the texture of the observed scene, the classification achieves good results. We present an overview of the achieved classification rates for all videos in figure 2. The *AUC* for *ROC* and *PR* are given as well as the F-scores. The $F_{0.5}$ and F_1 scores show that the algorithm has a high reliability. Especially for the *Alley*, *Alley Leveled*, *Narrow*, *Sidewalk*, *Sidewalk 2*, *Sidewalk Leveled* and *Street* videos we achieve a very high discrimination ratio, almost always above 90%. These videos have a fairly large stretch of accessible section with obstacles located on both sides in common as well as only very few obstructions separating the ground section. We also achieve good results on the *Car*, *Corridor*, *Fence*, *Hedge*, *Ladder*, *Passage* and *Sign* videos. Here, a variable amount of directly accessible ground section is available and it is often separated into several parts. Classification rates are still above 80% and support the initial assumption that the gradient classifier could deliver good results even under difficult circumstances. The *Bicycle*, *Flower-box*, *Pan*, *Railing*

Fig. 2: Per dataset video results: the *AUC* for the *ROC* and *PR* curves, F_β for $\beta = 0.5$ and $\beta = 1$ as well as per video accuracy (Acc.). $\bar{x}$ shows the average score calculated over all video scores.

Name	$\int$ROC	$\int$PR	$F_{0.5}$	F_1	Acc.
Alley	0.928	0.882	0.937	0.916	0.901
Alley L.	0.892	0.856	0.941	0.911	0.862
Bicycle	0.753	0.629	0.843	0.869	0.676
Car	0.850	0.679	0.763	0.739	0.851
Corridor	0.819	0.665	0.816	0.750	0.796
Fence	0.855	0.750	0.878	0.834	0.815
Flower-box	0.783	0.607	0.838	0.789	0.724
Hedge	0.836	0.827	0.882	0.872	0.814
Ladder	0.836	0.629	0.757	0.736	0.868
Narrow	0.958	0.924	0.922	0.928	0.929
Pan	0.759	0.548	0.843	0.861	0.650
Passage	0.850	0.733	0.889	0.821	0.805
Railing	0.760	0.626	0.842	0.852	0.696
Ramp	0.803	0.680	0.870	0.839	0.731
Ridge	0.854	0.622	0.230	0.304	0.199
Sidewalk	0.929	0.945	0.943	0.947	0.913
Sidewalk 2	0.947	0.914	0.913	0.912	0.904
Sidewalk L.	0.889	0.942	0.954	0.950	0.912
Sign	0.890	0.835	0.933	0.899	0.854
Street	0.940	0.885	0.919	0.904	0.917
$\bar{x}$	0.852	0.753	0.861	0.828	0.784

and *Ramp* videos show the algorithm's limits. Large parts of the ground section cannot be recognized. We observe the worst detection accuracy in the *Ridge* video. Much noise is created in the disparity map by the combination of grass partly covered with snow, which has very low texture information. Furthermore, this video has by far the smallest amount of visible ground section which causes the algorithm to detect more false positives in relation to other videos. We provide an example video in the supplemental material.

While the average accuracy over all videos is 78.4%, the accuracy over all frames is considerably higher at 91.74% when using optimal thresholds for each video. This difference can be explained by the varying length of the videos. In figure 3, we show a bad classification example, which is part of the *Pan* video. The system fails this situation and only a small percentage of the accessible ground section is correctly classified. The stereo reconstruction cannot deliver sufficient information and the classifier cannot deal with the created additional noise in the disparity map.

As we have shown, our approach allows for a fast and effective accessible section detection, even in crowded scenes. A major drawback however is our reliance on a good stereo reconstruction. While our system can perform in real time on a modern machine, we strongly depend on the quality of the reconstruction process. The stereo reconstruction we used for our evaluation is computationally expensive and prevents real time usability for user studies. However, using a *Kinect* a performance of 30 frames per second is achieved without putting heavy load on the used machine. We observe another deficit in border regions between accessible section and obstacles. Due to our chosen fixed kernel size, we fail to detect small obstacles and reduce our accuracy along the way. We could mitigate this effect through the implementation of varying kernel sizes depending on the observed situation and their necessity.

6. CONCLUSION

We presented a simple yet efficient method to determine the accessible section in front of a walking person. We use the depth information provided by, e.g., a stereo camera to calculate the ground plane and subsequently derive the section of the image that is not blocked by obstacles. In order to evaluate the quality of the proposed approach, we recorded a novel dataset. The dataset consists of 20 videos depicting urban scenes that were recorded using a hand held stereo camera rig. It contains realistic amounts of lighting variations, ego-motion, and scene variety in urban scenarios. We will make the software framework, its base modules, as well as the dataset publicly available.

As part of our future work, we plan to extend our work to monocular camera setups to allow for future application on off-the-shelf mobile phones. Furthermore, we intend to investigate how we can use haptic or auditory output modalities to communicate the information to visually impaired users.

7. REFERENCES

[1] S. Shoval, I. Ulrich, and J. Borenstein, "Navbelt and the guide-cane [obstacle-avoidance systems for the blind and visually impaired]," *IEEE Robot. Autom. Mag.*, vol. 10, no. 1, pp. 9–20, 2003.

[2] Sylvain Cardin, Daniel Thalmann, and Frederic Vexo, "Wearable obstacle detection system for visually impaired people," in *VR workshop on haptic and tactile perception of deformable objects*, 2005.

[3] J Coughlan and R Manduchi, "A mobile phone wayfinding system for visually impaired users," *Assistive technology research series*, vol. 25, pp. 849, 2009.

[4] Diansheng Chen, Wei Feng, Qiteng Zhao, Muhua Hu, and Tianmiao Wang, "An infrastructure-free indoor navigation system for blind people," *Intelligent Robotics and Applications*, pp. 552–561, 2012.

[5] Juan Manuel Saez Martinez, Francisco Escolano Ruiz, et al., "Stereo-based aerial obstacle detection for the visually impaired," in *ECCV Workshop on Computer Vision Applications for the Visually Impaired*, 2008.

(a) (b) (c)

Fig. 3: Here a low recognition score is caused by a failing of the stereo reconstruction algorithm. The images show the open area of a parking space (a), the problems of the disparity map with large holes and artifacts (b) and the failing accessible section detection (c).

[6] Chia-Hsiang Lee, Yu-Chi Su, and Liang-Gee Chen, "An intelligent depth-based obstacle detection system for visually-impaired aid applications," in *Image Analysis for Multimedia Interactive Services (WIAMIS), 2012 13th Int. Workshop on*, 2012.

[7] Young Hoon Lee and Gérard Medioni, "Rgb-d camera based navigation for the visually impaired," in *Proc. Robotic Science and Systems*, 2011.

[8] Stephen Se and Michael Brady, "Ground plane estimation, error analysis and applications," *Robotics and Autonomous Systems*, vol. 39, no. 2, pp. 59–71, 2002.

[9] Martin A Fischler and Robert C Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[10] Young Hoon, Lee Tung-sing Leung, and Gérard Medioni, "Real-time staircase detection from a wearable stereo system," 2012.

[11] P. Lombardi, M. Zanin, and S. Messelodi, "Unified stereovision for ground, road, and obstacle detection," in *Proc. Intelligent Vehicles Symposium*, 2005.

[12] R. Labayrade, D. Aubert, and J.-P. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation," in *Proc. Intelligent Vehicle Symposium*, 2002.

[13] C. Braillon, C. Pradalier, J.L. Crowley, and C. Laugier, "Real-time moving obstacle detection using optical flow models," in *Proc. Intelligent Vehicles Symposium*, 2006, pp. 466–471.

[14] Richard O Duda and Peter E Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[15] Berthold KP Horn and Brian G Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1, pp. 185–203, 1981.

[16] Dennis Mitzel and Bastian Leibe, "Close-Range Human Detection for Head-Mounted Cameras," *The British Machine Vision Association and Society for Pattern Recognition*, pp. 1–11, 2012.

[17] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "Moving obstacle detection in highly dynamic scenes," in *International Conference on Robotics and Automation*, may 2009, pp. 56–63.

[18] Andreas Ess, Konrad Schindler, Bastian Leibe, and Luc Van Gool, "Object detection and tracking for autonomous navigation in dynamic environments," *The International Journal of Robotics Research*, vol. 29, no. 14, pp. 1707–1725, 2010.

[19] Andreas Geiger, Martin Roser, and Raquel Urtasun, "Efficient large-scale stereo matching," *Computer Vision–ACCV 2010*, pp. 25–38, 2011.

[20] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, 2009, vol. 3.

[21] Nikolay Chumerin and Marc M Van Hulle, "Ground plane estimation based on dense stereo disparity," in *The Fifth International Conference on Neural Networks and artificial intelligence, Minsk, Belarus*, 2008, pp. 209–213.