

# Detection of Ascending Stairs using Stereo Vision

Hannes Harms<sup>1</sup>, Eike Rehder<sup>1</sup>, Tobias Schwarze<sup>1</sup> and Martin Lauer<sup>1</sup>

**Abstract**—Environment perception is an important task in computer vision for many applications in robotics. Especially for robots navigating through different levels of a building, stair detection constitutes an important perception task. In this paper, we propose a stair detection algorithm using range data. Firstly, we introduce a parameter, which describes local surface orientations w.r.t. a global reference. Secondly, a matched filter is used to detect relevant edges in the orientation data. Afterwards, line segments are determined using these edge data which are further used to estimate stairs. The proposed method is invariant against rotations of the sensor. We show that the system can handle typical outdoor stair types and outperforms the accuracy of state-of-the-art stair detection methods. Moreover, the method is used in real time to assist visually impaired people who wear the camera system on a helmet.

## I. INTRODUCTION

In recent years, 3d vision sensors are used in surface reconstruction [13], object detection [12], gesture recognition [21] and many other fields in robotics. The substantial progress made in the field of environment perception is widely applied to autonomously navigating robots and safety assistance systems for automobiles.

Presently, these advancements have received more interest in assistance systems for visually impaired people [1]. Blind people usually rely on feedback from a white cane, navigating very cautiously in environments that are unknown or in which steps and stairs occur. Especially in these situations, it is helpful for the visually impaired to provide additional information of the surrounding. Curbs and stairs represent hazardous situations in everyday indoor and outdoor environments.

In this paper, we present a detection system for ascending stairs to provide additional information for the visually impaired, e.g. the location and orientation of stairs and the dimensions of the steps. The main contribution of this paper is a rotational invariant 3d edge detection method, which is determined by a parameter, that considers local surface orientations w.r.t. a global height axis. Moreover, we present a stair model, which allows the detection of multiple steps and stairs from temporally tracked 3d line segments. The results demonstrate the accuracy of the proposed approach compared to other state of the art stair detection systems. The presented work is evaluated on a real-time capable system, that combines a stereo camera rig and an inertial measurement unit (IMU) which are mounted on a helmet.

<sup>1</sup>Hannes Harms, Eike Rehder, Tobias Schwarze and Martin Lauer are with Institute of Measurement and Control Systems, Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany {harms, eike.rehder, tobias.schwarze, martin.lauer}@kit.edu

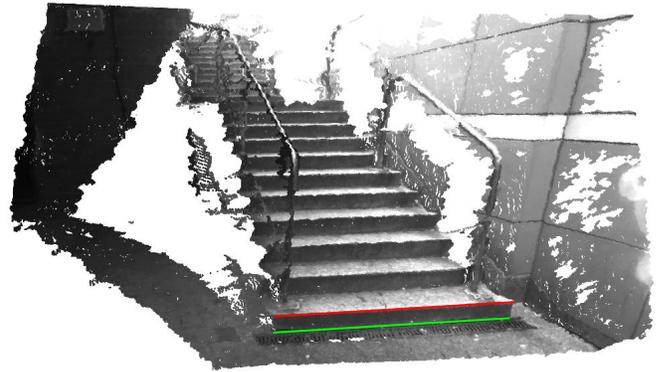


Fig. 1: Reconstructed 3d point cloud of a typical outdoor stair scene. The line segments of the first step are marked as concave (green) and convex (red).

This paper is structured as follows: Section II gives a brief overview of state of the art stair detection research. Section III introduces the novel stair detection method, describing all relevant processing steps in detail. In section IV the used hardware is specified. The experimental results of the introduced method are presented in section V. We close this paper with conclusions and an outlook.

## II. RELATED WORK

Detection of staircases has been focus of research in the last decades, especially in the field of mobile robotics and the assistance of visually impaired people. In this section, we will give a brief overview of state of the art stair detection methods. So far, two main approaches were proposed for stair modeling: plane and line extraction based methods. Line-based stair detection methods firstly detect convex and/or concave stair line segments to fit these line segments to a stair model afterwards (see fig. 1), whereas plane-extraction based methods consider stairs to be detected as a sequence of planar and vertical plane segments. Previously used sensor hardware varies from monocular cameras to 3d sensors like stereoscopic and RGB-D cameras or lidars.

Plane extraction based methods are commonly used with range sensors. Oehler et al. [16] group surface elements with similar aligned normals (*surfels*) from an RGB-D camera. The scene is further segmented by estimating planes into connected *surfels*. Pradeep et al. [19] compute point-wise surface normals from stereo camera data. Afterwards, tensor voting is used to calculate globally consistent normals and clustering is performed to fit planes onto the 3d point cloud.

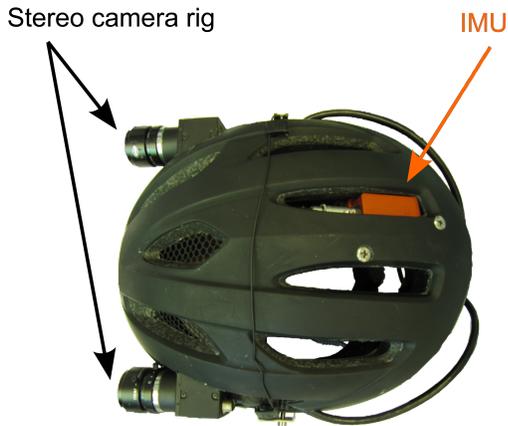


Fig. 2: Topview of the prototype hardware setup. The stereo camera rig is front directed (left in the image), the IMU is integrated in the helmet.

Osswald et al. [17] extract planes for steps and risers from 3d point clouds, which is assembled by tilting a 2d lidar mounted on a robot's head. Recently, Yus et al. [20] have proposed a method for RGB-D cameras which uses region growing on 3d point clouds, taking normal orientation and curvature information into account.

Line-based stair detection approaches usually extract edges from a monocular camera which are clustered in groups of concurrent line segments. The variation of intensities is used to differentiate between convex and concave line segments. Obviously, challenges are expected if shadows in the scene produce false detections or intensity gradients are low due to difficult lighting conditions.

A texture detection method based on Gabor filters was proposed by Se and Brady [22] using a monocular camera. Concave and convex line segments are distinguished by the variation of intensity. A combination of Gabor filters and fuzzy fusion phase grouping was presented by Zhong et al. [24]. Delmerico et al. [3] use an RGB-D camera to extract stair edges from depth images, fuse the edge observations and estimate a stair model. Approaches using a gray value image and additional 3d information (e.g. from a stereo camera rig or RGB-D setup) take depth steps in the 3d-point cloud into account to distinguish convex from concave line segments. A commonly used method for line detection is the Hough transform (HT) [6]. Lately, the authors of [14] combined the depth discontinuities with an edge detection from a single gray value image. This information is fused by a weighted HT which classifies the edge segments as concave or convex. Stereo artifacts proved to be the main challenge in this approach and could only partially be compensated by additional heuristics. Approaches using texture information rely on the existence of intensity gradients, which are not always present and are therefore another disadvantage.

Even though there is a huge diversity of different line-

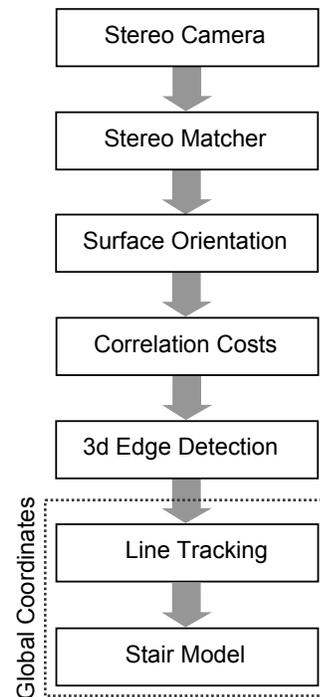


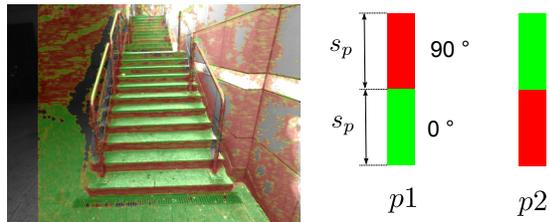
Fig. 3: Processing pipeline used in this paper.

based stair detection approaches, there is no approach to our best knowledge, that uses range information without texture information to estimate convex and concave line segments directly. Such an approach has the advantage to be portable to any other range sensor. In addition, drawbacks of approaches using texture information can be overcome, e.g. false positive detections from misleading shadows edges or missed detections in image regions with low intensity gradients due to difficult lighting conditions. Our approach meets the challenge of line detection from depth data and is introduced in the next section.

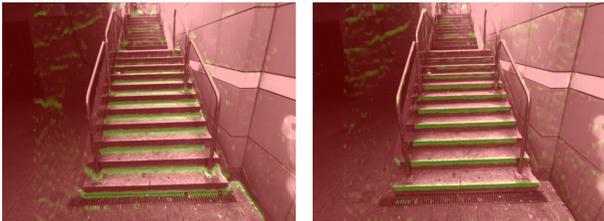
### III. STAIR DETECTION

This section introduces the line-based stair detection algorithm step by step, following the processing pipeline (see fig. 3). Our goal is to describe a stair as a set of steps, whereas each step consists of a pair of a convex and a concave line segment, see fig. 1. We estimate concave and convex line segments from depth data directly (i.e. from dense disparity maps), track these line segments over time and fit a stair model into the tracked line segments.

The used sensors in this work are a stereo camera system and an IMU, which are both calibrated intrinsically and extrinsically. Both sensors are mounted on a helmet (see fig. 2). Further information regarding the sensor hardware can be found in section IV. The two sensors are triggered, so that a pair of rectified gray value images and a global orientation of the helmet is delivered for every processing step. A state-of-the-art stereo matcher with sub-pixel accuracy is used to compute a dense disparity image [9] [7]. The dense disparity image is the input for the following processing steps.



(a) Surface normal orientation image  $I_\delta$  (b) Convolution pattern  $p1$  and  $p2$



(c) Cost image  $C_1$  (d) Cost image  $C_2$

Fig. 4: From orientation image  $I_\delta$  to correlation costs  $C_1$  and  $C_2$

### A. Surface Orientation

In a first step, we compute the surface normals in the sensor domain, as suggested by Badino et al. [2]. The pixel-wise estimated normals  $\mathbf{n}_{\text{disp}}$  in disparity space are transformed to world space  $\mathbf{n}_{\text{world}}$ , as explained in [8]. A parameter is determined for each pixel position, which describes the deviation angle  $\delta$  between the local surface normal  $\mathbf{n}_{\text{world}}$  and a global height axis  $\mathbf{h}_{\text{world}}$  in world coordinates. The deviation angle  $\delta$  is defined as

$$\delta = \arccos \frac{\mathbf{h}_{\text{world}} \cdot \hat{\mathbf{n}}_{\text{world}}}{\|\mathbf{h}_{\text{world}}\| \|\hat{\mathbf{n}}_{\text{world}}\|}. \quad (1)$$

Note that  $\delta$  is the dihedral angle, so that  $\delta \in [0^\circ, 90^\circ]$ . The global height axis  $\mathbf{h}_{\text{world}}$  can be determined from an independent measurement of an IMU (acceleration and magnetic sensors) or estimated by tracking the ground plane normal vector over time. The parameter  $\delta$  is invariant against rotations of the camera system because of the chosen global height axis reference. The output image  $I_\delta$  contains the deviation angles  $\delta$  for each pixel position  $(u,v)$ , see fig. 4a. For further processing steps,  $I_\delta$  is normalized ( $I_\delta \in [0, 1]$ ).

### B. Correlation Costs

The next processing step generates two cost images from  $I_\delta$ , that are later used to extract concave and convex line segments. Both cost images consist of pixel-wise cost values, which describe a transition from flat to vertical surfaces ( $C_1$ ) and vice versa ( $C_2$ ). In the following, the cost images are abbreviated by  $C_j$ ,  $j \in \{1, 2\}$ . The measurements  $C_j$  use correlation operations on  $I_\delta$ . The correlation masks  $p_j$  (see fig. 4b) are applied on  $I_\delta$  and can be expressed by the following column-wise correlation functions



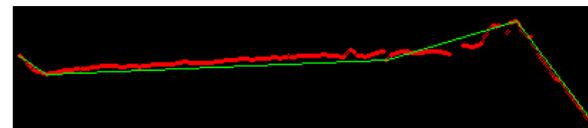
(a) Binary mask  $I_{bin_1}$  from  $C_1$  (b) Binary mask  $I_{bin_2}$  from  $C_2$  for concave edges for convex edges



(c) One segment of binary mask (ROI)



(d) Multiplication of binary mask segment with cost image. Black indicates high costs, white low costs.



(e) Red points stand for minimum points, green line segments are estimated by iterative end point fit method

Fig. 5: Line segment estimation

$$C_1(u, v) = 1 - \frac{1}{s_p} \left( \sum_{i=v-s_p}^v I_\delta(u, i) - \sum_{i=v}^{v+s_p} I_\delta(u, i) \right) \quad (2)$$

$$C_2(u, v) = 1 - \frac{1}{s_p} \left( \sum_{i=v}^{v+s_p} I_\delta(u, i) - \sum_{i=v-s_p}^v I_\delta(u, i) \right) \quad (3)$$

The parameter  $s_p$  in pixels is defined by the discrete pixel height estimated for each 3d point in world coordinates assuming a certain height  $h_m$  in meters (e.g.  $h_m = 0.2$  meters)

$$s_p(d) = \frac{d h_m}{b}, \quad (4)$$

where  $d$  represents the current disparity value at pixel position  $(u,v)$  and  $b$  the baseline in meters. The larger the scene depth, the smaller gets  $s_p$ . The correlation functions work similarly to a matched filter, see [15] for further details. The two output images  $C_j$  provide pixel-wise values, which describe the membership to a transition from flat to vertical surfaces and respectively. See fig. 4 for an example of  $C_j$ . The color green corresponds to low costs (i.e. the pattern fits), while red corresponds to high costs (i.e. the pattern does not fit).

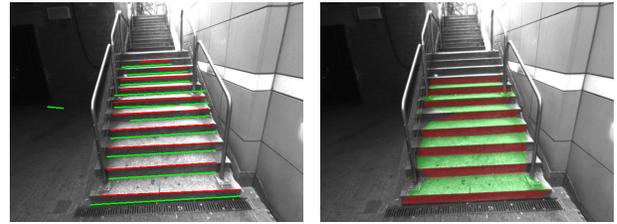
### C. 3d Edge Detection

This step provides relevant areas of  $C_j$ , which are used to estimate line segments. Both histograms  $h_{C_j}$  of the cost images  $C_j$  reveal a characteristic bimodal distribution of values. To achieve a differentiation of the maximum distributions for varying highpeaks, we maximize the intra-variance  $\sigma_w$  of  $h_{C_j}$  as proposed by Otsu et al. [18] and determine a binary image for each  $C_j$ . Connected regions in the binary image are clustered using a border following algorithm [23]. Assuming a minimum size  $S_e$  for relevant stair edge regions, two binary masks  $I_{bin_j}$  are extracted, which contain the region of interests (ROIs) for further stair edge estimation, see fig. 5a and 5b.

During the next step, line segments are estimated in the previously extracted ROIs. Each binary ROI is multiplied with the appropriate cost image  $C_j$ , see fig. 5d. For each resulting ROI, the shortest path (e.g. by using a Dijkstra graph solver [4]) describes the global optimal solution in  $C_j$ . Unfortunately, calculating a shortest path is computationally expensive and requires a predefined start and end point, which are not known because the ROI can be orientated arbitrarily. We approximate such a shortest path by rotating each ROI in the image plane, so that the principal component of each ROI is aligned with the image u-axis. Afterwards the column-wise minima are determined, as can be seen in fig. 5e. A direct solution to obtain stair line segments would be the estimation of a line for each ROI, using a least square estimator or M-estimator. The results of such a straight forward solution are inaccurate, due to infrastructure objects or noise effects (see fig. 5d) which appear in the same ROIs with stair line segments. Most stairs are composed of steps which can be described by line segments or a set of line segments. Thus, each minimum cost path can be represented by a set of line segments. To achieve that, an iterative end-point fit algorithm as introduced by Ramer et al. [5] is used. The result is shown in fig. 5e. The green line segments approximate the distribution of the red minimum points. Finally, the green line segments are refined by fitting a line trough the minimum points of each line segment using a least square estimator. The output of this processing step are line segments with convex and concave labels, see fig. 6a.

### D. Line Tracking

The convex and concave line segments are tracked over time and stored in a map which has a fixed 3d world coordinate system. The association step compensates the ego motion between two consecutive frames by visual odometry [7]. Line segments which are detected in a new frame are associated to the closest existing line segment in the map, assuming small ego motion estimation errors. To consider only stable line segment tracks (=tracks) for the stair detection, the tracks in the map have two states, valid and invalid. A track is considered valid if a minimum number of line segment detections  $n_{min}$  within the preceding time period  $T_p$  were associated to it, otherwise the track is invalid. The geometric parameters of a valid track are position, length



(a) Detected 3d edges

(b) Detected stair

Fig. 6: Detected 3d edges and fitted stair model

and orientation. The parameters of each track are filtered by averaging the parameter values from all associated line segments. New tracks are added to the map if an observed line segment from a new image is not associated to an existing track and fulfills a minimum distance  $d_{min}$  to all existing tracks. Tracks are maintained in the map if leaving the camera's viewing field, while tracks in the viewing field are erased without any associated line segments in the preceding time period  $T_p$ . In fig. 8e tracked convex and concave line segments are shown, the same track number is indicated by temporal consistent colors.

### E. Stair Model

The stair model is fitted from valid line segment tracks with convex and concave labels, see fig. 8e. In general, stairs can be described as a set of steps, whereas each step consists of a pair of a convex and a concave line segment. At first, single steps are modeled as a pair of a convex and a concave line segment. Fig. 1 shows such a pair of line segments, which we define as a step. Secondly, all detected steps are combined in consecutive sets, which represent stairs. For each step and stair detection step a minimal number of features is used which contain the height and orientation of the considered line segments and the shortest distance relative to each other.

The constraints to consider a pair of a convex and a concave line segment as a step are:

- the height difference is smaller than  $\Delta h_{step}$ ,
- the orientation difference of the line segments is smaller than  $\Delta o_{step}$  and
- if we can pair a line segment with several other line segments, we choose the line segment with the most appropriate dimensions and distance.

Two steps belong to the same stair if the following constraints are fulfilled:

- the height difference of the convex line segment of one step relative to the concave line segment of the other step is smaller than  $\Delta h_{stair}$ ,
- the orientation difference of these two line segments is smaller than  $\Delta o_{stair}$  and
- if we can pair a step with several other steps, we choose the step with the most appropriate dimensions and distance.

The result of such a stair modeling can be seen in fig. 6b. All detected steps are visualized as transparent red polygons,

Quantity	Ours	RGB-D [3]	TPRS [17]	SLG [17]
Height Error [cm]	$0.12 \pm 0.66$	$1.7 \pm 1.4$	$0.42 \pm 0.33$	$0.68 \pm 0.54$
Percent	0.8	8.9	6.0	9.7
Depth Error [cm]	$0.24 \pm 1.14$	$1.2 \pm 1.6$	$1.17 \pm 0.67$	$0.9 \pm 0.61$
Percent	0.85	4.2	6.5	5.0
Width Error [cm]	$19.43 \pm 10.8$	$17.3 \pm 12.8$	$3.4 \pm 1.95$	$2.25 \pm 1.97$
Percent	10.7	16.5	5.7	3.8
Pitch Error [°]	$0.09 \pm 1.28$	$2.3 \pm 1.9$		
Plane Error (parallel)			$2.22 \pm 2.17$	$1.14 \pm 1.13$
Plane Error (90°)			$4.97 \pm 2.13$	$3.12 \pm 1.47$

TABLE I: Step modeling errors comparing the average error  $\pm$  standard deviation (AVG $\pm$ STD) of our method to an RGB-D based method [3], and the lidar based methods Scan-Line Grouping (SLG), and Two-Point Random Sampling (TPRS) [17].

steps assigned to the same stair are connected by transparent green polygons. Note that all inconvenient 3d line segments shown in fig. 6a are rejected as step candidates, due to insufficient geometric connections to other line segments or being defined as invalid by line tracking.

#### IV. HARDWARE SETUP

The proposed stair detection algorithm is intended to assist visually impaired people. A prototype of the system is currently realized by a helmet, on which the sensors are mounted, see fig. 2. A stereo camera rig is mounted on the helmet with two Point Grey Flea2 cameras. The stereo cameras have a resolution of 640 x 480 pixels, a focal length of 360 pixels and a baseline of 18.3 cm.

An inertial measurement unit (IMU) is as well mounted on the helmet (MTi-300-AHRS from XSENS). In the running system, the IMU is currently used to deliver a global reference height axis, as described in section III-A. All computations are done on a notebook, which is carried in a backpack.

Dense disparity images are computed using semi-global block matching, as introduced by Hirschmüller et al. [9]. The normal vectors in disparity space are computed using integral images, see [11] and [10] for more information. The run time for each step of the processing pipeline can be found in table II.

Processing step	Run time in [ms]
Rectification and stereo matching	38
Visual odometry	35
Normal computation	40
Surface orientation	5
Correlation costs	5.5
3d edge detection	20
Line tracking	0.11
Stair model	0.04

TABLE II: Run times of the proposed method.

By pipelining the three first processing steps which are computationally most expensive our system runs at approximately 10 Hz.

Our framework is realized in C++ and uses a standard notebook with an 2.67 GHz dual core CPU, on which all algorithms are computed.

#### V. RESULTS

The line tracking and stair detection step are initialized by the following parameters:

$$\begin{aligned}
 n_{min} &= 3 \text{ frames} \\
 T_p &= 15 \text{ frames} \\
 d_{min} &= 0.2 \text{ meters} \\
 \Delta o_{step} &= \Delta o_{stair} = 10^\circ \\
 \Delta h_{step} &= 0.25 \text{ meters} \\
 \Delta h_{stair} &= 0.05 \text{ meters}
 \end{aligned}$$

For robustness reasons, the input line segments for tracking are restricted to observed line segments which can be assigned to steps in the set of currently observed concave and convex line segments. These constraints proved to be sufficient to model stairs from detected sets of convex and concave line segments. The parameters can be easily adopted to work for other stair types, e.g. spiral stairs or stairs considering steps with larger heights.

The evaluation of the proposed method considers ascending stairs which are approached and traversed. The stair dimensions are estimated from the temporally filtered global stair model. There is currently no publicly available stair detection benchmark or data set which can be used to compare the results on the same sequence. We apply our approach on a typical outdoor stair and determine the estimated step height, depth, width and pitch by the average error and standard deviation. Table I compares the results of the proposed stereo vision based method with three state of the art stair detection methods, one using an RGB-D camera [3] and two using lidar data [17]. The estimation of the step height, depth and pitch outperforms the accuracy of state of the art approaches, see table I. The step width is mainly underestimated due to occlusions from the handrails or splitted line segments resulting from depth discontinuities, see fig. 8d. Note that the stair landing is detected as connecting area between the two stairs. The estimated stair landing depth of 222 cm is close to the ground truth depth of 223 cm. A reconstructed complete stair model can be found in figure 7. The stair estimation of the proposed method uses only noisy disparity data from a stereo camera as compared to other approaches using more accurate depth data from RGB-D cameras [3] or laser scanners [17].

Result images of the evaluated outdoor stair are presented in fig. 8. Each column shows the same stair scene for one

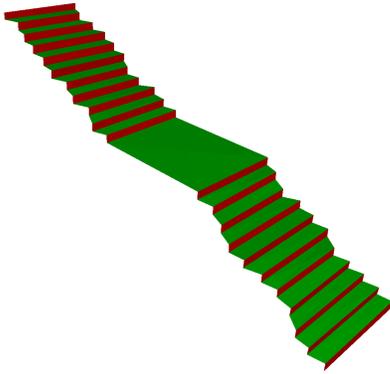


Fig. 7: Result of estimated global 3d stair model

time step. Visualized are the orientation images  $I_\delta$ , the cost images  $C_1$  and  $C_2$ , the detected concave and convex line segments (3d edges), the valid concave and convex line segment tracks and the estimated stair model as introduced in section III from the tracked line segments. Detected steps in fig. 8f are visualized as transparent red polygons, green polygons connect steps belonging to the same stair. Note that all inconvenient 3d edges shown in fig. 8d are rejected as step candidates. The stair model using tracked line segments compensates missing and partly detected steps. Especially in the near field, the line tracking approach merges the line segments very accurately. All mis-associations made during traversing the stair end up in invalid line segment tracks. All steps that are close to the camera system are merged correctly into a stair. The detected stair in column two of fig. 8f is separated properly by the stair landing. For further example stairs and visualizations, we reference to the attached video file of this contribution.

Our method detects stairs that are orientated arbitrarily in the image, as long as the concave and convex line segments of the stair are visible.

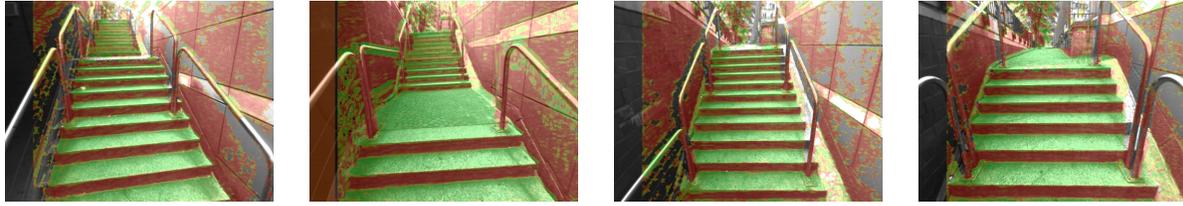
## VI. CONCLUSION AND OUTLOOK

In this contribution, we introduced a stair detection method that is based on range data. A matched filter uses surface normal orientations to fit convex and concave line segments in the environment. These line segments are tracked over time and are combined to step segments, which are furthermore merged to stairs. The experimental results show that our method achieves robust detections, as can be seen in the previous result section. Moreover a highly accurate model of the traversed stair is estimated, which outperforms state of the art stair detection methods. Disadvantages of line based stair detection approaches based on texture information, like false detections at pedestrian crosswalks or shadow regions do not occur.

Further work is planned to replace the global reference height axis from the IMU by a tracked surface normal vector of the ground plane. The detection of descending stairs using a matched filter for depth steps is currently in work as well as a localization method, which uses the proposed line segment tracking to realize a 3d SLAM.

## REFERENCES

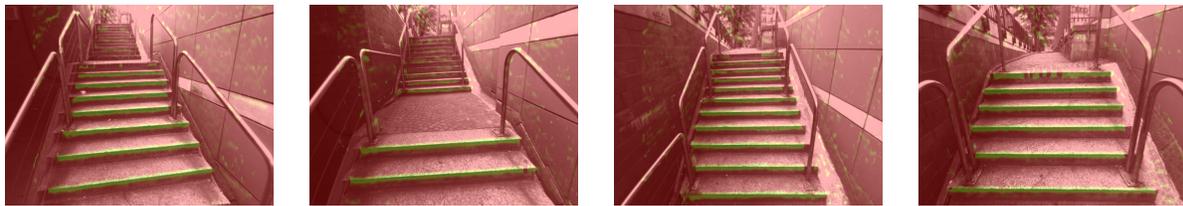
- [1] *ECCV 2014 Workshops - Assistive Computer Vision and Robotics*, 2014.
- [2] H. Badino, D. Huber, Y. Park, and T. Kanade. Fast and accurate computation of surface normals from range images. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3084–3091. IEEE, 2011.
- [3] J. A. Delmerico, D. Baran, P. David, J. Ryde, and J. J. Corso. Ascending stairway modeling from dense depth imagery for traversability analysis. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2283–2290. IEEE, 2013.
- [4] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- [5] D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, 1973.
- [6] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.
- [7] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *Computer Vision—ACCV 2010*, pages 25–38. Springer, 2011.
- [8] H. Harms, J. Beck, J. Ziegler, and C. Stiller. Accuracy analysis of surface normal reconstruction in stereo vision. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 730–736. IEEE, 2014.
- [9] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):328–341, 2008.
- [10] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke. Real-time plane segmentation using rgb-d cameras. In *RoboCup 2011: Robot Soccer World Cup XV*, pages 306–317. Springer, 2012.
- [11] S. Holzer, R. B. Rusu, M. Dixon, S. Gedikli, and N. Navab. Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2684–2689. IEEE, 2012.
- [12] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 1999.
- [13] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM Siggraph Computer Graphics*, volume 21, pages 163–169. ACM, 1987.
- [14] X. Lu and R. Manduchi. Detection and localization of curbs and stairways using stereo vision. In *IEEE International conference on Robotics and Automation*, volume 4, page 4648. Citeseer, 2005.
- [15] D. O. North. An analysis of the factors which determine signal/noise discrimination in pulsed-carrier systems. *Proceedings of the IEEE*, 51(7):1016–1027, 1963.
- [16] B. Oehler, J. Stueckler, J. Welle, D. Schulz, and S. Behnke. Efficient multi-resolution plane segmentation of 3d point clouds. In *Intelligent Robotics and Applications*, pages 145–156. Springer, 2011.
- [17] S. Obwald, J.-S. Gutmann, A. Hornung, and M. Benezit. From 3d point clouds to climbing stairs: A comparison of plane segmentation approaches for humanoids. In *Humanoid Robots, 2011 11th IEEE-RAS International Conference on*, pages 93–98. IEEE, 2011.
- [18] N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.
- [19] V. Pradeep, G. Medioni, J. Weiland, et al. Piecewise planar modeling for step detection using stereo vision. In *Workshop on computer vision applications for the visually impaired*, 2008.
- [20] A. Prez-Yus, G. Lopez-Nicols, and J. J. Guerrero. Detection and modelling of staircases using a wearable depth sensor. In *ECCV 2014 Workshops - Assistive Computer Vision and Robotics*, 2014.
- [21] Z. Ren, J. Yuan, and Z. Zhang. Robust hand gesture recognition based on finger-earth mover’s distance with a commodity depth camera. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1093–1096. ACM, 2011.
- [22] S. Se and M. Brady. Vision-based detection of staircases. In *Fourth Asian Conference on Computer Vision ACCV*, volume 1, pages 535–540, 2000.
- [23] S. Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.
- [24] C. Zhong, Y. Zhuang, and W. Wang. Stairway detection using gabor filter and fpg. In *Soft Computing and Pattern Recognition (SoCPaR), 2011 International Conference of*, pages 578–582. IEEE, 2011.



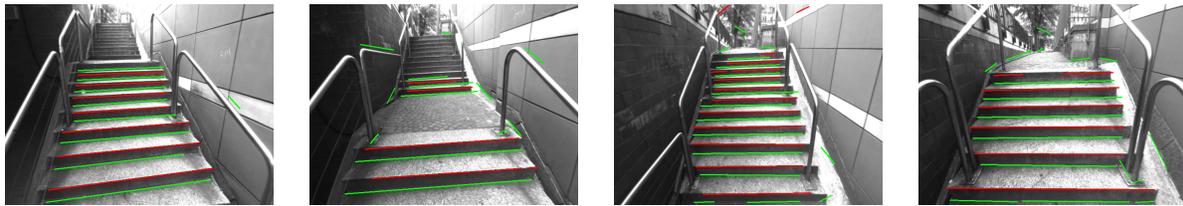
(a) Orientation images  $I_\delta$



(b) Cost images  $C_1$



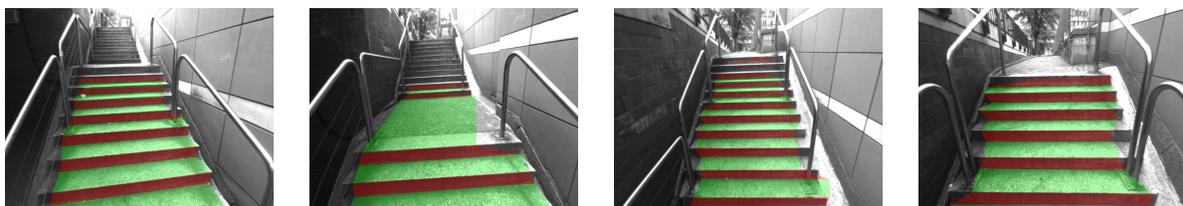
(c) Cost images  $C_2$



(d) Detected 3d edges. Concave edges are visualized in green, convex edges in red.



(e) Tracked line segments, same color means same track number.



(f) Estimated stair model. Red polygons are detected steps, green polygons connect steps belonging to the same stair.

Fig. 8: Results of proposed stair detection approach