Visual Shoreline Detection for Blind and Partially Sighted People

Daniel Koester, Tobias Allgeyer, and Rainer Stiefelhagen

Karlsruhe Institute of Technology, Karlsruhe, Germany
{daniel.koester,tobias.allgeyer,rainer.stiefelhagen}@kit.edu

Abstract. Currently existing navigation and guidance systems do not properly address special guidance aides, such as the widely used white cane. Therefore, we propose a novel shoreline location system that detects and tracks possible shorelines from a user's perspective in an urban scenario. Our approach uses three dimensional scene information acquired from a stereo camera and can potentially inform a user of available shorelines as well as obstacles that are blocking an otherwise clear shoreline path, and thus help in shorelining. We evaluate two different algorithmic approaches on two different datasets, showing promising results. We aim to improve a user's scene understanding by providing relevant scene information and to help in the creation of a mental map of nearby guidance tasks. This can be especially helpful in reaching the next available shoreline in yet unknown locations, *e.g.*, at an intersection or a drive-way. Also, knowledge of available shorelines can be integrated into routing and guidance systems and vice versa.

Keywords: Assistive System · Orientation & Mobility · Shorelines

1 Introduction

Today's navigation systems have become ubiquitous, as every smartphone capable of running *Google Maps*[1], it's accompanying *Maps* smartphone application, or other available navigation software, literally puts this capability into everyone's pocket. It was just recently that some of these added specialized pedestrian modes–*Google Maps* did so only in 2015–and provided a huge improvement over the so far road based routing. However, almost none of the existing systems address any special requirements, such as those of people affected by cerebral palsy, people with walking aids, or blind and partially sighted people. Therefore, to this modern day and age, it is common for blind and partially sighted people to mostly rely on other, more traditional means: the traditional skill of white cane usage–a skill that has to be learned through *Orientation and Mobility Training*–as well as guide dogs, or less common, echolocation.

In recent years, mostly through the creation of crowdfunded projects, especially the *Open Street Map*[2] project, innovation in those fields has hugely

[1] https://google.com/maps
[2] https://openstreetmap.org

K. Miesenberger and G. Kouroupetroglou (Eds.): ICCHP 2018, LNCS 10897, pp. 1–9, 2018.
https://doi.org/10.1007/978-3-319-94274-2_18

improved for researchers and hobbyists alike, as many more data sources are now publicly available [1] and their quality continues to increase over time [2]. Using such available data sources, especially geolocation meta data, *e.g.*, location and availability of accessible pedestrian traffic lights, zebra crossings, ramps, footbridges and many more, the creation of adapted routing algorithms has become feasible. Also, it is now possible to create routes that integrate the usage of assistive systems into low-level routing decisions. Recently, Koester *et al.* [3] have investigated such routing on a shoreline-level detail, in order to provide routes for blind and partially sighted people. They try to adhere to specific requirements and include learned white cane usage patterns by integrating shorelines into the routing process.

In this work, we try to visually detect and track shorelines from a first person view approach, using three dimensional, stereo reconstruction data provided by a body worn camera. A few examples of our shoreline detection can be seen in varying situations in Fig. 1.

First, we calculate and track the main vanishing point in our image and determine a region of interest to search for shorelines, usually below the vanishing point to left and right. Those shorelines are mostly haptic edges of some kind, *e.g.*, between the sidewalk and building walls, walkway side curbs or walkway borders, and are visible as such in a stereo reconstruction. We have developed two slightly different variations to track these, both provide a candidate for each frame. Then we track and average our shoreline candidates over multiple frames and report the most promising one(s).

Our proposed system is able to inform users of available close by shorelines that point towards their current walking direction and can help to locate the next available shoreline from afar. This is especially useful after road crossings or in urban areas, where buildings are separated by driveways and the next shoreline might be hard to find, or navigate to in a straightforward manner. The system proves especially helpful in unknown locations, where a lot of white cane users would most likely not feel comfortable to explore the terrain by themselves. Furthermore, having additional shoreline availability information can greatly improve a user's scene layout understanding in general as well as the personally perceived safety when navigating in busy urban areas.

2 Related Work

While navigation systems have become widespread, they usually don't adapt to special needs, or often require expensive specialized hardware, as stated by Csapó *et al.* [4] in a survey of routing applications for blind and partially sighted people. Except for the recent shoreline level routing proposed by Koester *et al.* [3], existing routing approaches do not acknowledge the special requirements created by shorelining and the specific requirements this technique creates for such systems when it comes to low level decisions.

First person view approaches for navigation and orientation of blind and partially sighted people have so far mostly focused on very specific and limited

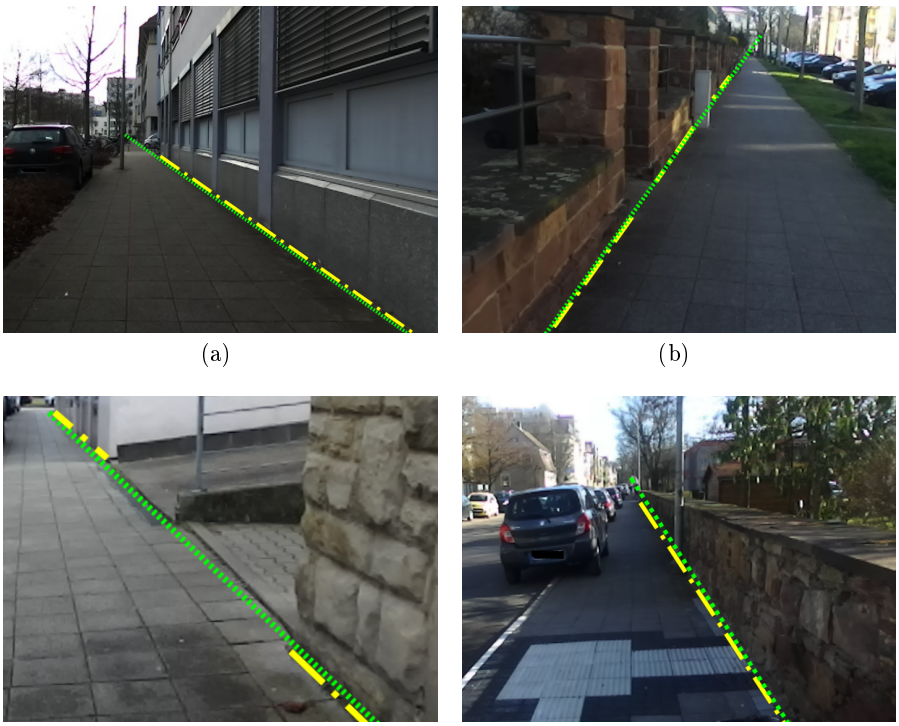

Fig. 1. Four example images, where the yellow dashed line shows the reference label, while the green dotted line is our detection. (a) is the full image of a building's facade, while (b)-(d) are magnified cutouts for improved visibility, where (b) shows a stone-fence, (c) a gap in the shoreline due to a driveway and (d) a stone wall. All four show our algorithm is capable of very accurately detecting shorelines in varying situations.

street crossing modalities, such as zebra crossing detection by Se [5], Ahmetovic *et al.* [6,7,8], or Ivanchenko *et al.* [9,10], and traffic lights, for example by Ivanchenko *et al.* [11]. A survey by Elmannai *et al.* [12] provides a comprehensive overview of general wearable assistive navigation and orientation devices, and discusses benefits and limitations in detail.

To the best of our knowledge, no existing work has so far tried to detect and track inner or outer shorelines for white cane related usage of blind and partially sighted people, using a visual approach from a first person viewpoint. The most similar work has been created by Coughlan *et al.* [13,14] and Ivanchenko *et al.* [15], who presented systems for curb and shoreline detection for blind and partially sighted wheelchair users, in order to prevent them from hazards and obstacles. Although their systems were not intended for active white cane usage, their motivation and intention is quite similar to ours.

3 Methodology

Our algorithm consists of two main components: a vanishing point detection and the actual shoreline detection. Furthermore, we use a simple tracking approach for both steps to improve and stabilize their detection over time. Currently, our systems assumes the existence of a vertical segment next to the ground-plane, *i.e.*, a building's wall or a fence adjacent to the pavement and we assume an urban area in a *Manhattan Style World*. Whether such a wall is generally available can also be gleaned from additional information, for example the shoreline-level routing provided by Koester *et al.* [3], although, the required data and accuracy is not guaranteed for in the underlying *Open Street Map*, as the quality might vary greatly between locations.

3.1 Visual Odometry

Visual odometry data is readily provided by the *Stereolabs ZED*[3] camera for our own dataset and we use *LIBVISO2*[4] [16] for the *Flowerbox*[5] dataset (as it does not provide odometry). We estimate the user's current direction of movement from that visual odometry data by averaging the location differences of the last 10 camera positions. This provides us with an estimated direction of possible future shoreline candidates, as those have to be somewhat aligned to the walking direction. Since we assume a *Manhattan Style World*, this direction often also aligns with the vanishing point, which we also try to detect.

3.2 Vanishing Points

Vanishing points are detected similar to Wu *et al.* [17], albeit with some major differences. First, we use a different line detection algorithm, *EDLines* [18], to search for all straight line segments, but discard all almost horizontal and vertical segments, as these usually don't contribute to vanishing points – at least in a *Manhattan Style World* almost all edges are either horizontal/vertical or point straight towards a vanishing point. Similar to Wu *et al.* [17], we then weigh line segments based on their direction, *i.e.*, directions closer to 45° are better, and length, *i.e.*, where longer is better. Finally, we use RANSAC [19] to determine candidates and decide on the point with the highest accumulated score.

These points are tracked over time (one per frame) in world coordinates by converting them into spherical coordinates and using a two-dimensional decaying accumulator array over a sphere's surface, where the maximum of all cells yields the averaged vanishing point coordinates. Those are then projected into the 2D image, which stabilizes the vanishing point in the always moving camera image with respect to the person's own ego-motion.

[3] https://www.stereolabs.com/zed/
[4] http://cvlibs.net/software/libviso/
[5] https://cvhci.anthropomatik.kit.edu/~dkoester/data/flowerbox.zip

3.3 Shorelines

Using the vanishing point, we determine a region of interest in the disparity image or the point cloud from the stereo reconstruction. This region is located below the vanishing point, usually to the left or right of it, where one would expect to find inner or outer shorelines, *i.e.*, sidewalk curbs or building walls.

Since the *ZED* depth camera directly provides a point cloud and surface normals, we can use those normals to distinguish between ground (upright surfaces) and walls (vertical surfaces), by comparing them with our movement vector and assuming a somewhat upright held camera (a constraint that is usually satisfied by body worn cameras). We use an implementation [20] of an optimized RANSAC modification by Chum *et al.* [21] to generate surfaces from those identified ground and wall pixels. The intersection of ground and wall surface then represents our shoreline candidate for the current frame.

When we do not have point clouds and surface normals (the *Flowerbox* dataset does not provide them), we rely only on the disparity image. We then filter the above mentioned region in the image using a median of a Gaussian to smooth noise and outliers in the depth data. After calculating gradients in X and Y direction of the disparity image, we can separate horizontal surfaces, *e.g.*, ground pixels, from vertical surfaces, *e.g.*, walls or curbs, using fixed thresholds. A morphological *opening* operation closes small holes and discards some more outliers. Finally, we determine shoreline candidate points by combining border regions from horizontal and vertical surfaces in regions where these borders overlap. We then transform these points into 3D space and use RANSAC to calculate our final shoreline candidate.

In both cases, we then merge consecutive shoreline candidates, *i.e.*, from consecutive images in a similar location, by using a confidence based weighted combination of these candidates. We keep a list of candidates and track their confidences over time, updating them each time a candidate is found in the currently evaluated frame. Keeping only valid candidates over time, we return the average of all remaining candidates. This tracking approach also stabilizes the shoreline and improves its distance and orientation with respect to the actually existing shoreline.

4 Experimental Evaluation

We have evaluated our approach on suitable parts of the *Flowerbox* dataset, as this dataset provides us with calibrated stereo images, taken from a first person viewpoint in an urban setting. We only consider videos where shorelines are clearly visible, *i.e.*, building walls or curbs and label the visible shorelines. To calculate an angular error and the minimal distance of our detected shoreline to the label, we convert our 2D labels into a 3D line by projecting the individual pixels of the labeled line in the 2D image into 3D coordinates and then use RANSAC to determine the best fitting line in 3D world coordinates.

This allows us to directly calculate the three dimensional angular error, *i.e.*, the angle between the two lines viewed as vectors from the same origin. The

Table 1. Shoreline detection results for 3 videos of the *FlowerBox* dataset. Angles (median $\tilde{\Theta}$, mean $\bar{\Theta}$, standard deviation δ_Θ) are given in degrees and distances (median $\tilde{d}$, mean $\bar{d}$, standard deviation δ_d) are in cm, calculated using transformed real world points and (shore-)lines.

	$\tilde{\Theta}$	$\bar{\Theta}$	δ_Θ	$\tilde{d}$	$\bar{d}$	δ_d
sidewalk	$0.6°$	$3.8°$	$9.5°$	$2.2\,cm$	$6.8\,cm$	$18.2\,cm$
sidewalk-2	$1.4°$	$3.9°$	$9.5°$	$1.7\,cm$	$3.3\,cm$	$5.7\,cm$
sidewalk-leveled	$0.4°$	$1.1°$	$3.9°$	$2.4\,cm$	$2.4\,cm$	$1.3\,cm$

Table 2. Shoreline detection results for 6 videos of our own data for our disparity image based algorithm (same algorithm as used for *Flowerbox*). Angles and distances are given as in Table 1. Some digits had to be omitted for space reasons.

	frame-wise detection						averaged detection					
	$\tilde{\Theta}$	$\bar{\Theta}$	δ_Θ	$\tilde{d}$	$\bar{d}$	δ_d	$\tilde{\Theta}$	$\bar{\Theta}$	δ_Θ	$\tilde{d}$	$\bar{d}$	δ_d
I	5.8	8.0	11.1	0.28	0.40	0.43	7.2	8.3	5.6	0.40	0.55	0.43
II	13	14	9.9	0.10	0.10	0.06	13	14	8.9	0.12	0.21	0.36
III	2.3	4.4	5.2	0.19	0.21	0.14	1.8	2.8	3.0	0.22	0.26	0.20
IV	2.1	8.2	12	0.12	0.16	0.19	3.2	7.2	10.9	0.13	0.19	0.23
V	2.6	3.0	1.9	0.12	0.12	0.07	3.0	8.4	18.3	0.13	0.43	0.86
VI	2.2	2.5	1.4	0.16	0.16	0.06	2.5	4.2	3.5	0.19	0.46	0.53

Table 3. Shoreline detection results for 6 videos of our own data for our point cloud based algorithm, using surface normales. Angles and distances are given as in Table 1.

	frame-wise detection						averaged detection					
	$\tilde{\Theta}$	$\bar{\Theta}$	δ_Θ	$\tilde{d}$	$\bar{d}$	δ_d	$\tilde{\Theta}$	$\bar{\Theta}$	δ_Θ	$\tilde{d}$	$\bar{d}$	δ_d
I	2.0	6.2	14.2	0.07	0.15	0.26	1.5	2.9	3.4	0.12	0.20	0.24
II	3.0	5.4	7.7	0.05	0.09	0.20	1.6	2.0	1.8	0.06	0.07	0.08
III	1.5	4.6	9.2	0.04	0.06	0.13	1.1	2.0	2.8	0.03	0.03	0.02
III	1.5	4.6	9.2	0.04	0.06	0.13	1.1	2.0	2.8	0.03	0.03	0.02
IV	4.2	7.7	14.1	0.03	0.04	0.06	2.4	3.0	1.8	0.02	0.06	0.14
V	1.2	1.7	2.4	0.03	0.04	0.07	1.1	1.3	0.8	0.02	0.03	0.11
VI	1.4	4.1	8.4	0.04	0.06	0.17	1.3	1.6	1.0	0.02	0.05	0.04

Table 4. Average values over all videos of Table 2 and 3, comparing the disparity image based algorithm (DI) with the point cloud based algorithm (PCL). Angles and distances are given as in Table 1.

	frame-wise detection						averaged detection					
	$\tilde{\Theta}$	$\bar{\Theta}$	δ_Θ	$\tilde{d}$	$\bar{d}$	δ_d	$\tilde{\Theta}$	$\bar{\Theta}$	δ_Θ	$\tilde{d}$	$\bar{d}$	δ_d
DI	4.6	6.7	7.0	0.16	0.19	0.16	5.1	7.4	8.4	0.20	0.35	0.43
PCL	2.2	4.9	9.3	0.04	0.08	0.15	1.5	2.1	1.9	0.04	0.07	0.10

minimal distance is calculated by determining the three dimensional distance of our detected shoreline candidate to each point on the labeled line, as detection and label are almost always a set of skewed lines that do not intersect.

Evaluation results for the most viable three of the *FlowerBox* videos are given in Table 1. Furthermore, we evaluate our approach on our own data recorded with a *Stereolabs ZED* camera and compare both versions of our algorithm, the disparity image based algorithm in Table 2 as well as the point cloud and surface normal based algorithm in Table 3. We show that, especially for the point cloud based version, the averaging and tracking over multiple frames helps a lot to reduce errors. Finally, we compare both algorithms in Table 4, where the point cloud based method greatly outperforms the disparity image based algorithm.

5 Conclusion

We demonstrate an algorithm to detect shorelines from a first person view. To the best of our knowledge, no such system exists so far for blind and partially sighted people for use with the white cane. Our proposed shoreline detection algorithm, which relies on vanishing points and edges it can detect and track, achieves very promising results and works very accurate for the tested scenarios. This approach could potentially be used to aide in shorelining, *i.e.*, locating and navigating along shorelines, a technique commonly used by blind and partially sighted people. However, some adaption to less strong visible shorelines, such as walkway borders, is required to increase its usefulness and further evaluation is needed. Finally, we plan to test our approach in a user study to determine it's usefulness and integrate it as a part of a guidance system for blind and partially sighted people that is capable of providing fine grained orientation and guidance information to the user.

Acknowledgements

This work has been partially funded by the *Bundesministerium für Bildung und Forschung (BMBF)* under grant no. 16SV7609.

References

1. Goodchild, M.F.: Citizens as sensors: the world of volunteered geography. GeoJ. 69(4), 211–221 (2007)
2. Zielstra, D., Zipf, A.: Quantitative Studies on the Data Quality of OpenStreetMap in Germany, In: Proceedings of GIScience (2010)
3. Koester, D., Awiszus, M., Stiefelhagen, R.: Mind the Gap: Virtual Shorelines for Blind and Partially Sighted People. In: IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 1443–1451 (2017)
4. Csapó, Á., Wersényi, G., Nagy, H., Stockman, T.: A Survey of Assistive Technologies and Applications for Blind Users on Mobile Platforms: A Review and Foundation for Research. In: Journal on Multimodal User Interfaces **9**(4), pp. 275–286 (2015)

5. Se, S.: Zebra-crossing detection for the partially sighted. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 211-–217 (2000)
6. Ahmetovic, D., Bernareggi, C., Mascetti, S.: Zebralocalizer: identification and localization of pedestrian crossings. In: Proceedings of the International Conference on Human Computer Interaction with Mobile Devices and Services, pp. 275-–286 (2011)
7. Ahmetovic, D., Bernareggi, C., Gerino, A., Mascetti, S.: ZebraRecognizer: efficient and precise localization of pedestrian crossings. In: 22nd International Conference on Pattern Recognition (ICPR), pp. 2566-–2571 (2014)
8. Ahmetovic, D., Manduchi, R., Coughlan, J., Mascetti, S.: Mind your crossings: Mining GIS imagery for crosswalk localization. ACM Trans. Access. Compu. **9**(4), pp. 11:1–11:25 (2017)
9. Ivanchenko, V., Coughlan, J., Shen, H.: Detecting and locating crosswalks using a camera phone. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (2008)
10. Ivanchenko, V., Coughlan, J.M., Shen, H.: Crosswatch: a camera phone system for orienting visually impaired pedestrians at traffic intersections. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2008. LNCS, vol. 5105, pp. 1122-–1128. Springer, Heidelberg (2008)
11. Ivanchenko, V., Coughlan, J., Shen, H.: Real-Time Walk Light Detection with a Mobile Phone. In: Miesenberger, K., Klaus J., Zagler W., Karshmer A. (eds) ICCHP 2010. LNCS vol. 6180, pp. 229–234. Springer, Heidelberg (2010)
12. Elmannai, W., Elleithy, K.: Sensor-Based Assistive Devices for Visually-Impaired People: Current Status, Challenges, and Future Directions. In: Sensors **17**(3), pp. 565 (2017)
13. Coughlan, J., Manduchi, R., Shen, H.: Computer Vision-Based Terrain Sensors for Blind Wheelchair Users. In: Miesenberger, K., Klaus J., Zagler W., Karshmer A. (eds) ICCHP 2006. LNCS vol. 4061, pp. 1294–1297, Springer, Heidelberg (2006)
14. Coughlan, J., Shen, H.: Terrain Analysis for Blind Wheelchair Users: Computer Vision Algorithms for Finding Curbs and other Negative Obstacles. In: CVHI (2007)
15. Ivanchenko, V., Coughlan, J., Gerrey, W., Shen, H.: Computer vision-based clear path guidance for blind wheelchair users.In: Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 291–292 (2008)
16. Geiger, A., Ziegler, J., Stiller, C.: StereoScan: Dense 3D Reconstruction in Real-time. In: ntelligent Vehicles Symposium (IV), pp. 963–968 (2011)
17. Wu Z., Fu W., Xue, R., Wang, W.: A Novel Line Space Voting Method for Vanishing-Point Detection of General Road Images. In: Sensors **16**(7):948 (2016)
18. Akinlar, C., Topal, C.: EDLines: Real-Time Segment Detection by Edge Drawing. In: 18th IEEE International Conference on Image Processing (ICIP), pp. 2837–2840 (2011)
19. Fischler, M., Bolles, R.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. In: Commun. ACM **24**(6), pp. 381–395 (1981)
20. Raguram, R., Chum, O., Pollefeys, M., Matas, J., Frahm, J.: USAC: A Universal Framework for Random Sample Consensus. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, **35**(8), pp. 2022–2038 (2013)
21. Chum, O., Matas, J.: Optimal Randomized RANSAC. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, **30**(8), pp. 1472–1482 (2008)