

Camera-based On-Road Detections for the Visually Impaired

Judith Jakob^{1,2}, József Tick³

¹ Óbuda University Budapest, Doctoral School of Applied Informatics and Applied Mathematics, Bécsi út 96/b, 1034 Budapest, Hungary

² Furtwangen University of Applied Sciences, Faculty of Computer Science, Robert-Gerwig-Platz 1, 78120 Furtwangen, Germany
e-mail: jaj@hs-furtwangen.de

³ Óbuda University Budapest, Institute of Applied Informatics, John von Neumann Faculty of Informatics, Bécsi út 96/b, 1034 Budapest, Hungary
e-mail: tick@uni-obuda.hu

Abstract: Herein, we research the possibilities of assisting visually impaired pedestrians moving in traffic situations by using camera-based detection of relevant objects in their immediate surroundings. Therefore, we use and adapt algorithms from the field of driver assistance. We present a road background segmentation based on watersheds, whose results are used as input for the presented crosswalk and lane detection algorithms. The crosswalk detection is based on the application of two 1D mean filters and the lane detection on local computations of the EDF (Edge Distribution Function). In our evaluation, the described algorithms achieved good hit rates of 99.87% (road segmentation), 98.64% (crosswalk detection), and 97.89% (lane detection).

Keywords: ADAS; Driver Assistance; Visually Impaired; Road Segmentation; Crosswalk Detection; Lane Detection; Computer Vision

1 Introduction

According to [6], in 2015, worldwide 36 million people were estimated to be blind and 216.6 million people were estimated to have a moderate to severe visual impairment. Since 1990, the numbers of both groups have increased: in comparison to 1990, there was a rise of 17.5% in blind people and 35.5% in people with moderate to severe visual impairment.

In their study from 2001, Duckett and Pratt underline the importance of mobility for the visually impaired: “The lack of adequate transport was described as resulting in many visually impaired people living in isolation. Transport was felt

to be the key to visually impaired people fulfilling their potential and playing an active role in society,” [10].

Therefore, we research possibilities of assisting visually impaired people by using camera-based object detection in traffic situations in order to increase their independent mobility. As the amount of research in the field of Advanced Driver Assistance Systems (ADAS) is much higher in comparison with Assistive Systems for the Visually Impaired (ASVI), our research focuses on transferring algorithms and concepts from ADAS to ASVI.

1.1 Previous Work

In [19], we motivate the need for a transfer concept from ADAS to ASVI and present a plan for an ASVI consisting of a smartphone app and a cloud service. A camera as well as earphones to provide text-to-speech output are connected to the smartphone; expensive image processing calculations are exported to the cloud service. Although the system will not be built in the near future, we keep this system in mind as reference system during our research. Furthermore, [19] contains first findings regarding the transfer concept and a short description of a preliminary version of the crosswalk detection algorithm that we describe in detail in this article.

In order to gather the needs of visually impaired people in traffic situations, we conducted qualitative interviews with experts and members of the target group using Witzel’s method of problem-centered interviews [34]. The results of the expert interviews are presented in [18] and [20]; a common evaluation of both interview types is not published yet. The evaluation of the interviews is presented in the form of tables, adapted from Sommerville’s software engineering book [30], describing the six identified traffic scenarios: (1) General orientation, (2) navigating to an address, (3) crossing a road, (4) obstacle avoidance, (5) boarding a bus, and (6) at the train station. For each traffic scenario, we recorded the vision use cases that could help visually impaired people in the according scenario. We then formed the overlap with vision use cases that are of interest in both considered fields, ADAS and ASVI. These are the use cases, we have to consider in our research: (1) Lane detection, (2) crosswalk detection, (3) traffic sign detection, (4) traffic light (state) detection, (5) (driving) vehicle detection, (6) obstacle detection, and (7) bicycle detection. Additionally, we described a preliminary version of the lane detection algorithm presented explicitly in this article in [20].

For the evaluation of the algorithms that are developed in the course of our research, comparable sequences from driver and pedestrian perspective covering the seven identified use cases are needed. While there are many publicly available data sets from driver perspective, e.g. the KITTI data set [14] and the German Traffic Sign Detection Benchmark [16], there are few data sets from pedestrian perspective and no data sets with comparable material from both perspectives.

Therefore, we created the CoPeD (Comparable Pedestrian Driver) data set for traffic scenarios [21]. It is licensed under the Creative Commons Attribution 4.0 International License¹ and hosted publicly². The data set is divided into four categories: (1) Lanes, (2) crossings (incl. traffic lights and crosswalks), (3) obstacles (incl. vehicles and bicycles), and (4) traffic signs with relevance to pedestrians.

1.2 Presented Work

The next step towards a transfer concept from ADAS to ASVI is the examination of the seven identified use cases with relevance in both fields concerning their possibilities of adaptation from ADAS to ASVI.

The additional consideration of a road background segmentation makes it possible to run some detection algorithms on only a subset of the original image. Crosswalk and lane detection can be executed on the road image, whereas traffic sign and traffic light detection are carried out on the background part of the image. Obstacles, including vehicles and bikes, can occur in both parts of the image and are therefore run on the complete image. Figure 1 illustrates the procedure. This article focuses on the middle path of the diagram in Figure 1, meaning that we propose a road background segmentation and solutions for the “on-road” detections crosswalks and lanes. Thereby, the segmentation result is used as a preprocessing step for the different detection algorithms. This means that when integrated into an assistive system, the segmentation result is not communicated to the user unlike the result of the requested detection algorithm.

The following chapter first gives an overview concerning related work for the topics camera-based ASVI as well as road background segmentation, crosswalk detection, and lane detection for ADAS and ASVI. Afterwards, we present our proposed ASVI solutions for these topics, each based on procedures from ADAS. The proposed algorithms are then evaluated on sequences from the CoPeD data set [21]. Finally, Chapter 4 outlines our future work.

2 Related Work

This chapter first gives an overview of camera-based assistance for the visually impaired. Then, related work of the three topics that have to be considered, namely road background segmentation, crosswalk, and lane detection, is cited. For each topic, we explain which ADAS procedures we chose for the adaptations presented in the following Chapter 3.

¹ <https://creativecommons.org/licenses/by/4.0/>

² <http://dataset.informatik.hs-furtwangen.de/>

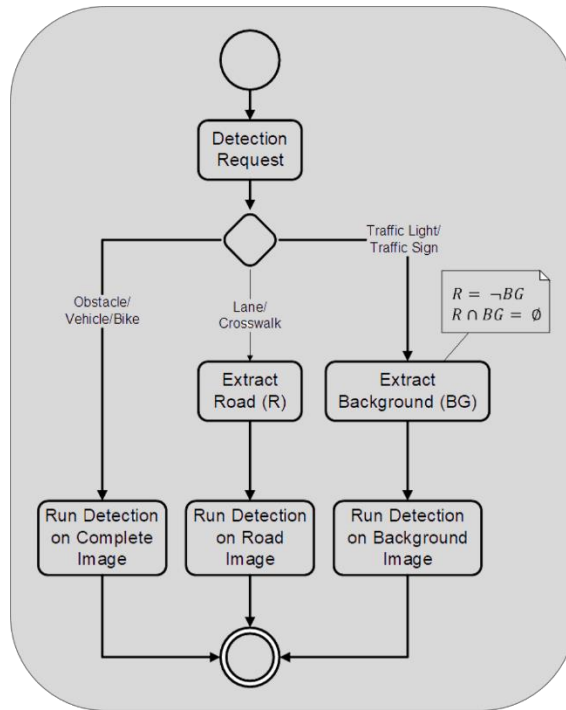


Figure 1

Partition of detections with road background segmentation

2.1 Camera-based Assistance for the Visually Impaired

We describe the state of the art of camera-based assistance for the visually impaired in [17]. The developed systems cover a wide range: In the case of sufficient remaining eyesight, camera images are represented in a suitable way (e.g. [32]). While other systems focus on the interpretation of image content in certain situations (e.g. [29]), Sensory Substitution Devices (SSD) offer a comprehensive perception of the environment by transferring the sense of sight to a different sense, e.g. hearing [7] or tactile sense [1].

In general, ASVI consist of a capture, an image processing, and an output unit [17]. The research presented in this paper focuses on the processing step and does at this stage not take the output - the way of communicating the results with the user - into account.

2.2 Road Background Segmentation

In the 1990s, road segmentation for driver assistance using morphological watersheds was introduced [5]. The algorithm uses markers situated in front of the

car; hence, it cannot be used without adaptations on data from pedestrian perspective.

From about 2008 onward, Alvarez et al. proposed road segmentation based on illumination-invariant models [2, 3, 4]. Their procedure poses two problems when used from pedestrian perspective: They use seeds in front of the car and a camera calibration is needed. The latter makes it difficult to use on any smartphone which is the platform an assistive system for the visually impaired is most likely to use.

Other algorithms are based on neural networks (e.g. [12]), support vector machines (e.g. [31]), or deep learning (e.g. [25]). These machine learning approaches are too complex and costly to be the first step in the image processing chain of an assistive system for the visually impaired.

Therefore, a road segmentation from pedestrian perspective based on Beucher et al.'s work using watersheds [5] is introduced in this article.

2.3 Crosswalk Detection

The adapted algorithm presented in the following chapter is based on the work of Choi et al. [9]. They introduce a combined detection of crosswalks and traffic lights. Their crosswalk detection is based on a 1-D mean filter in horizontal direction. Another example of an algorithm using the fact that crosswalks are horizontal structures from driver's perspective is described in [15]. Haseloff and Kummert apply Fourier and Hough Transform and thus make use of the bipolarity and straight lines which characterize crosswalks. Zhai et al. propose a crosswalk detection based on MSER and ERANSAC [37].

Furthermore, there is some research concerning crosswalk detection in the area of ASVI. The algorithms described in [28] and [33] are based on parallel lines that are extracted by Hough Transform. Cheng et al. [8] extract the bright crosswalk stripes by adaptive thresholding. They address challenging scenarios, such as partial occlusion, low contrast and distant crosswalks, and different illuminations. In addition, they offer an extensive literature review on crosswalk detection algorithms.

2.4 Lane Detection

In ASVI, the goal of lane detection is to help with general orientation by giving information about the course of the road, meaning indicating if the road goes on straight ahead or takes a left respectively right turn. On the other hand, in ADAS, lane detection is used to warn the driver if they risk to depart the lane. These lane departure warning systems consist of three steps, namely lane detection, lane tracking, and communication with the driver [24].

For ASVI, only the first step is of importance. According to [24], single frame lane detection is generally composed of four steps: (1) Image acquisition and preprocessing, (2) edge detection, (3) stripe identification by Hough Transform or Edge Distribution Function (EDF), (4) line fitting.

As the Hough Transform is not suitable to detect even slight curves, we concentrate our research on using the EDF which is the histogram of the gradient magnitude with respect to the corresponding edge angle. The work described in this paper is based on the lane detection using EDF presented by Lee [23]. After computing the region of interest with the help of the vanishing point, edge extraction as well as EDF construction are carried out. Based on an EDF analysis, Lee identifies if the car is safely within the lane or in risk of departing the lane boundaries.

3 Proposed ASVI Solutions

In the following, we describe our proposed solutions for road background segmentation as well as crosswalk and lane detection. Each reported procedure is based on known solutions from ADAS. In the end, we present an evaluation of all three procedures on a subset of the CoPeD data set [21].

In previous publications, we focused on the differences between the underlying ADAS algorithms and the proposed adaptations. This article, however, concentrates on the mathematical details of the proposed adaptations.

Table 1 shows the input and output variables for each procedure. It can be seen that road background segmentation is the foundation for the other two algorithms as the output from road background segmentation is used as input for crosswalk and lane detection.

Table 1

Input and output variables for road background segmentation, crosswalk detection, and lane detection

	Road Background Segmentation	Crosswalk Detection	Lane Detection
Input	I : RGB Input Image.	I, R	I, R
Output	R : Binary image; road pixels are white. BG : Binary image; background pixels are white ($BG = \neg R$).	CW : Binary image; crosswalk pixels are white.	$Text$: Indicates if the road is straight or takes a right/left turn, provided any lanes are detected.

In the complete chapter, (x, y) refers to a pixel position in a given image of size $N \times M$.

3.1 Road Background Segmentation

The presented algorithm is based on Beucher et al.'s work [5] and consists of five steps: After the watershed computation (1), properties of the according catchment basins are determined (2). With the help of this information, the mosaic image is built by assigning to each pixel the mean gray value of the catchment basin it belongs to (3). Afterwards, we merge adjacent catchment basins if their gradient is below a threshold (4). Finally, a decision is made which uniform region from the merged mosaic corresponds to the road part of the image and a morphological post-processing of this region is carried out (5).

Contrary to [5], we do not use a morphological gradient image as input of the watershed computation in order to reduce the number of catchment basins. To achieve this goal, we set some catchment basins to zero, depending on their mean value and mean saturation. Additionally, Beucher et al. use seeds in front of the car to expand the road from which is not possible in ASVI. To replace this procedure, we developed step (5), namely decision and morphological postprocessing.

(1) Watershed Computation

Watersheds are based on a topological interpretation of gray value images where the pixels, as spatial coordinates, are plotted against their intensity values. Local maxima are defined as watershed lines that separate the image into different regions. The inner parts of these regions are called catchment basins. In our implementation, we use the according method from Matlab's Image Processing and Computer Vision Toolbox.

In the following, we consider the notations:

I : RGB input image, smoothed with a Gaussian Filter.

$gray$: Gray value version of I .

s : s-channel (saturation) from I converted into HSV space.

L : Watershed image of $gray$. Every pixel has an assigned number k according to the catchment basin it belongs to. L is set to zero for watershed lines.

N : Number of catchment basins.

(2) Properties of the Catchment Basins

For every catchment basin k , $1 \leq k \leq N$, we compute the mean intensity and mean saturation value:

$$\overline{gray}(k) = \sum_{\{(x,y)|L(x,y)=k\}} \frac{gray(x,y)}{\#\{(x,y)|L(x,y)=k\}}$$

$$\bar{s}(k) = \sum_{\{(x,y)|L(x,y)=k\}} \frac{s(x,y)}{\#\{(x,y)|L(x,y)=k\}}$$

Thereby, # stands for the number of elements in a set.

(3) Mosaic Image

For $L(x, y) = k$, the mosaic image is then defined as

$$mosaic(x, y) = \begin{cases} \overline{gray}(k), & \text{if } k \neq 0 \wedge \bar{s}(k) < th_s \wedge th_1 < \overline{gray}(k) < th_2 \\ 0, & \text{else} \end{cases}$$

Depending on the thresholds th_s , th_1 , and th_2 , the value of the mosaic image is either set to the mean gray value or to zero.

If $mosaic(x, y)$ is set to zero, we simultaneously set $L(x, y)$ to zero. This means that the value of each pixel is set to the mean gray value of the corresponding catchment basin, provided mean gray value and mean saturation of the basin are in ranges that make them possible parts of the road.

(4) Merged Mosaic Image

In this step, adjacent catchment basins are merged, if the gradient between them falls below a threshold th_{gray} . Let k and l be the identification numbers of adjacent catchment basins. Their gradient is then defined as:

$$\nabla(k, l) = | \overline{gray}(k) - \overline{gray}(l) |$$

If it holds that $\nabla(k, l) < th_{gray}$, we update $\overline{gray}(k)$ and $\overline{gray}(l)$ to their mean value

$$\overline{gray}(k) = \overline{gray}(l) = \frac{\overline{gray}(k) + \overline{gray}(l)}{2}$$

and the mosaic image to the merged mosaic image

$$mosaic(x, y) = \begin{cases} \overline{gray}(k), & \text{if } (L(x, y) = k \vee L(x, y) = l) \wedge \nabla(k, l) < th_{gray} \\ mosaic(x, y), & \text{else} \end{cases}$$

Furthermore, we set $L(x, y) = k$ for all $\{(x, y)|L(x, y) = l\}$.

We repeat this process for all adjacent catchment basins.

(5) Decision and Morphological Postprocessing

Let max_i be the gray value that occurs most often in the merged mosaic image. We then define the binary road image R as:

$$R(x, y) = \begin{cases} 1, & \text{if } mosaic(x, y) = max_i \\ 0, & \text{else} \end{cases}$$

We perform morphological postprocessing by applying filling, opening, largest component, and closing on R .

The background is then determined as the complement of R :

$$BR = \neg R$$

For stability reasons, we use the union of the last n frames as a tracking step. Hence, the road of the n -th frame in a sequence is defined as:

$$R_n = \bigcup_{i=0}^{n-1} R_{n-i}$$

In tests, $n = 10$ led to good results.

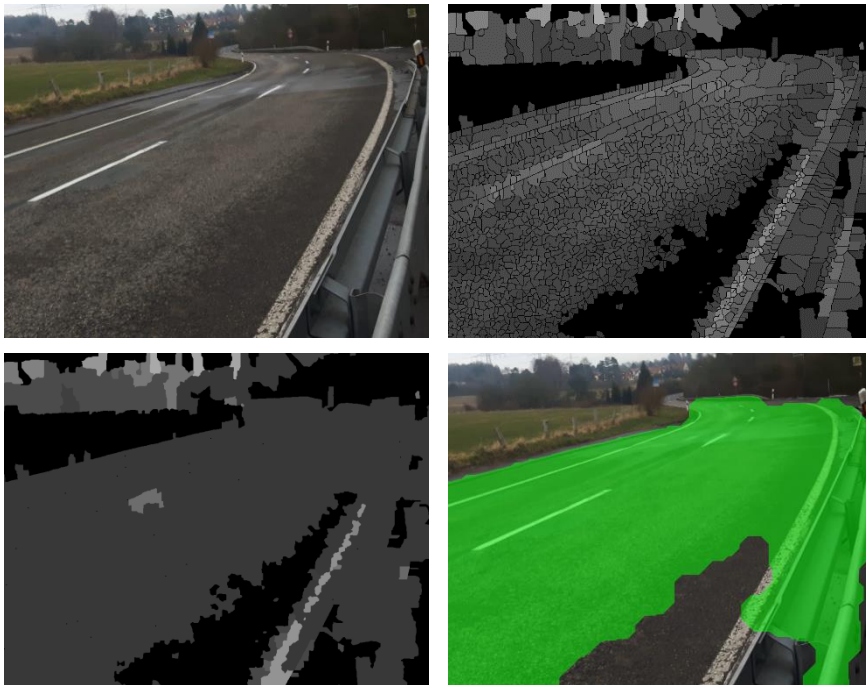


Figure 2

Intermediate steps of road background segmentation (cropped images):

Original, mosaic, merged mosaic, and result

Figure 2 shows intermediate steps of this algorithm for an example image. Even though, the road is not completely extracted, the segmentation counts as successful, because there are enough road detail to determine the road path. Furthermore, it is important that the bus stop sign (right side) was not extracted as road.

3.2 Crosswalk Detection

The presented algorithm is based on Choi et al.'s work [9] and preliminary descriptions can be found in [19] and [21]. It consists of four steps: After preprocessing and computing the Region of Interest (ROI) (1), horizontal and vertical 1-D mean filters are applied (2). The differences between original and filtered images are the foundation for binarization and morphological postprocessing (3). Finally, the two resulting masks are combined via a bitwise or operation and a decision is made if the remaining pixels form a crosswalk or not (4). According to Table 1, the original image as well as the extracted road from the road segmentation algorithm are the input variables for this algorithm.

Choi et al. [9] developed their algorithm for an ADAS containing a tilted front camera so that the resulting images show only the road, but no background. Therefore, no ROI was needed in [9] and we developed the before presented road background segmentation as ROI. Furthermore, we added a vertical filter in order to be able to detect crosswalks from every angle by combining it with the horizontal filter. As we consider two filters, our handling of the masks differs from the one in [9]. Finally, Choi et al. [9] gave no detailed description of their postprocessing and decision step so that we had to develop our own version.

(1) Preprocessing and ROI

As preprocessing, we apply a Gaussian filter for noise reduction. Alternatively, we use the already smoothed RGB image I from the road segmentation algorithm as input.

Let *gray* be the gray value version of I . The ROI is defined by the before computed road R . We therefore set the background pixels in *gray* to zero and keep the values belonging to the road:

$$gray(x, y) = \begin{cases} gray(x, y), & \text{if } R(x, y) = 1 \\ 0, & \text{else} \end{cases}$$

(2) 1-D Mean Filters

Choi et al. [15] propose a horizontal 1-D mean filter to detect crosswalks, because from driver's perspective crosswalks are horizontal structures. For pedestrians, however, crosswalks appear in every possible angle. Therefore, we use an additional 1-D mean filter in vertical direction. By combining the two filters, we are able to detect crosswalks occurring in any angle.

In order to respect the ROI, pixels having the value zero are excluded from the computation. With that, we get the result for filtering in horizontal respectively vertical direction as G_x and G_y :

$$G_x(x, y) = \sum_{k=-s}^s \frac{\text{gray}(x+k, y)}{\#\{(x+k, y) \mid -s \leq k \leq s \wedge \text{gray}(x+k, y) \neq 0\}}$$

$$G_y(x, y) = \sum_{l=-s}^s \frac{\text{gray}(x, y+l)}{\#\{(x, y+l) \mid -s \leq l \leq s \wedge \text{gray}(x, y+l) \neq 0\}}$$

The size of the filter is in both cases $(2 \cdot s + 1)$.

(3) Binarization and Morphological Postprocessing

We compute the differences between the original image gray and the filtered results G_x and G_y as D_x and D_y :

$$D_x(x, y) = |\text{gray}(x, y) - G_x(x, y)|$$

$$D_y(x, y) = |\text{gray}(x, y) - G_y(x, y)|$$

By applying a threshold th , we get the masks M_x and M_y :

$$M_x(x, y) = \begin{cases} 1, & \text{if } D_x(x, y) > th \\ 0, & \text{else} \end{cases}$$

$$M_y(x, y) = \begin{cases} 1, & \text{if } D_y(x, y) > th \\ 0, & \text{else} \end{cases}$$

Afterwards, we perform closing followed by opening with a larger structuring element on both masks independently.

(4) Bitwise or Operation and Decision

Before combining the two masks, we delete components that cover less than 1.5 % of the image. We then unite the two masks to one mask M by applying a pixelwise or operation:

$$M(x, y) = M_x(x, y) \vee M_y(x, y)$$

The largest component CW of M is a crosswalk candidate, provided it also exceeds the 1.5 % mark. The decision is made based on two values: The extent of CW indicating the relative number of pixels that are set inside a box surrounding CW and the ratio between minor and major axis length of CW .

If both, ratio and extent, are higher than thresholds th_r and th_e , CW is considered a crosswalk. Otherwise, no crosswalk is present and CW is set to zero.

Figure 3 shows intermediate results of the crosswalk detection algorithm for an example image. The figure only shows intermediate steps for vertical direction, because the crosswalk in this example is a vertical structure. Therefore, the horizontal mask is zero and the vertical mask is identical with the result.

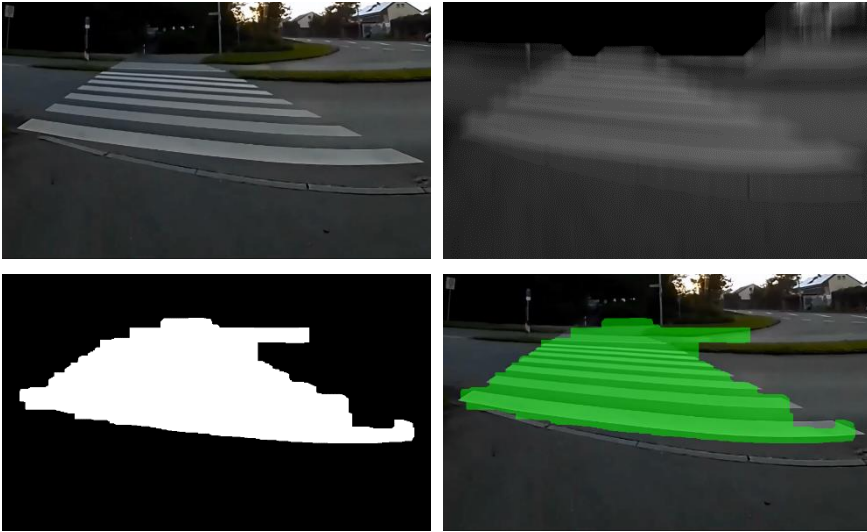


Figure 3

Intermediate steps of the presented crosswalk detection (cropped images):
Original, vertical filter, vertical mask, and result

3.3 Lane Detection

The presented algorithm is based on Lee's work [23] and a short description was published in [20]. Six subsequent steps are carried out: After preprocessing and computing the ROI based on the road background segmentation (1), we divide the ROI into a total of eight subimages (2). For every subimage, we compute (3) and analyze (4) the EDF, resulting in an angle for every subimage. Afterwards, the angles are interpolated (5) and the course of the road ahead is decided (6) based on the concavity of the interpolated function.

Lee's ROI computation [23] is based on the vanishing point. Because we cannot use this procedure in ASVI, we use the before extracted road as ROI. In contrast, EDF computation and analysis were mostly taken from [23]. The partition into subimages and the decision step were newly developed by us. As the purposes of lane detection differ for ADAS and ASVI, there are no according steps in [23]. Interpolation by linear parabolic fitting is carried out similarly to [24], but was adapted to the presented procedure.

(1) Preprocessing and ROI

This step is analogous to the before presented crosswalk detection algorithm. Additionally, the first row of the road R containing a non-zero entry is computed as min_y .

(2) Partition into Subimages

We divide the image from the bottom of the image to min_y into eight parts. From bottom to top, the first two subimages consist of one fourth of the available rows, the next two of one eighth, and the last four of one sixteenth.

The following two steps are carried out for each of the eight subimages. This will be indicated with the index j , where $j = 1$ refers to the subimage at the bottom which is the one closest to the user.

(3) EDF Computation

We compute the gradient $G_j(x, y) = (G_{j_x}(x, y), G_{j_y}(x, y))^T, j = 1, \dots, 8$, by convolving the image with the Sobel masks:

$$H_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

$$H_y = H_x^T$$

The magnitude $M_j(x, y)$ is then defined as

$$M_j(x, y) = \sqrt{G_{j_x}^2(x, y) + G_{j_y}^2(x, y)}, j = 1, \dots, 8$$

and the angle $\alpha(x, y)$ is computed as

$$\alpha(x, y) = \tan^{-1} \left(\frac{G_{j_y}(x, y)}{G_{j_x}(x, y)} \right), j = 1, \dots, 8,$$

so that the result is in the range $[1, 180]$. The degree values are rounded to integers.

Before determining the EDF, we compute the 97%-quantile of M and set the values below it to zero. The EDF is then defined as:

$$EDF_j(i) = \sum_{i=\alpha(x,y)} M(x, y), i = 1, \dots, 180, j = 1, \dots, 8$$

(4) EDF Analysis

First, we smooth $EDF_j, j = 1, \dots, 8$, with a 1D-filter of size 15. Afterwards, we compute the relative value of the highest peak $p_j \in [0, 1], j = 1, \dots, 8$ and its position $\theta_j \in [1, 180], j = 1, \dots, 8$. In case of no occurring peak, we set $p_j = 0$ and $\theta_j = 180$.

(5) Interpolation: Linear Parabolic Fitting

For interpolation, we consider the lane in the first two subimages from the bottom, the ones closest to the user, as linear and the remaining lane as parabolic. In order to be able to interpolate independently from image size, we normalize the values to $[0,1]$. Furthermore, we change the axes, meaning that the x -values are defined according to the size computed in step (2). With that, we get the interpolation function

$$f(x) = \begin{cases} a \cdot (x - 0.5) + b, & \text{if } x \leq 0.5 \\ a \cdot (x - 0.5) + b + c \cdot (x - 0.5)^2, & \text{else} \end{cases}$$

and the x -values $x = (0, 0.25, 0.5, 0.625, 0.75, 0.8125, 0.875, 0.9375, 1)^T$.

The according y -values are computed by determining a line with angle θ_j :

$$y_1 = 0$$

$$y_j = y_{j-1} + (x_j - x_{j-1}) \cdot \tan(\theta_{j-1} - 90), j = 2, \dots, 9$$

It is necessary to subtract 90 from the angle, because we rotated the axes. After computing, the y -values, we normalize them to $[0,1]$.

To get the interpolation function f , we compute the mean square error solution of the following overdetermined system of equations:

$$\begin{pmatrix} x_1 - 0.5 & 1 & 0 \\ x_2 - 0.5 & 1 & 0 \\ x_3 - 0.5 & 1 & 0 \\ x_4 - 0.5 & 1 & (x_4 - 0.5)^2 \\ x_5 - 0.5 & 1 & (x_5 - 0.5)^2 \\ x_6 - 0.5 & 1 & (x_6 - 0.5)^2 \\ x_7 - 0.5 & 1 & (x_7 - 0.5)^2 \\ x_8 - 0.5 & 1 & (x_8 - 0.5)^2 \\ x_9 - 0.5 & 1 & (x_9 - 0.5)^2 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \end{pmatrix}$$

(6) Decision

There are four possible outcomes for the algorithm: The input image does not contain lanes, the road goes straight ahead, or it takes a right or left turn.

First, we determine if the image contains lanes by checking if the mean value of the highest peaks $\bar{p} = \frac{1}{8} \sum_{j=1}^8 p_j$ is within a certain range $[th_{p_{low}}, th_{p_{high}}]$.

This is based on the idea that there is a characteristic amount of pixels belonging to the lane boundaries that indicate the direction. In tests, these pixels made up between 1.4% and 3% of the subimage.

If \bar{p} lies outside the specified range, the output text is set to ‘No lane detected’. Otherwise, we check the course of the road by examining the parameter c from the interpolation function and the variance of the θ -values $\bar{\theta} = \frac{1}{7} \cdot \sum_{j=1}^8 (\theta_j - \bar{\theta})$, where $\bar{\theta}$ is the mean θ value. In general, a negative c value means that the parabola is concave down and the road takes a right turn; accordingly, a positive value of c means that the parabola is concave up and the road takes a left turn. The higher the absolute value of c , the tighter the curve of the road; the lower the absolute value of c , the straighter the road. At the same time, the variance of the angles within the image is higher when the curve is tighter. With this knowledge, we set up a four-step process to determine the course of the road depending on several thresholds:

- (1) If $|c| < th_{c_{low}}$, the output text is ‘Straight ahead’.
- (2) If $th_{c_{low}} \leq |c| < th_{c_{mid}}$, the output is determined as:

$$\text{text} = \begin{cases} \text{'Straight ahead'}, & \text{if } \bar{\theta} < th_{var_{high}} \\ \text{'Left turn'}, & \text{if } \bar{\theta} \geq th_{var_{high}} \wedge c > 0 \\ \text{'Right turn'}, & \text{else} \end{cases}$$

- (3) If $th_{c_{mid}} \leq |c| < th_{c_{high}}$, the output is determined as:

$$\text{text} = \begin{cases} \text{'Straight ahead'}, & \text{if } \bar{\theta} < th_{var_{low}} \\ \text{'Left turn'}, & \text{if } \bar{\theta} \geq th_{var_{low}} \wedge c > 0 \\ \text{'Right turn'}, & \text{else} \end{cases}$$

- (4) If $|c| \geq th_{c_{high}}$, the output is determined as:

$$\text{text} = \begin{cases} \text{'Left turn'}, & \text{if } c > 0 \\ \text{'Right turn'}, & \text{else} \end{cases}$$

To increase the robustness, we compute the mean values of \bar{p} and $\bar{\theta}$ for the last 15 frames and set the output text to the one occurring the most in the last 15 frames.

Figure 4 shows the interpolated function for an example image. In general, the interpolated function does not match the course of the road exactly, but is a good approximation and has identical concavity.

3.4 Evaluation

The three before described algorithms were implemented in *Matlab Version R2017b* using the *Image Processing and Computer Vision Toolbox*. The output of the road segmentation was used as input for the ROI computation of crosswalk and lane detection.

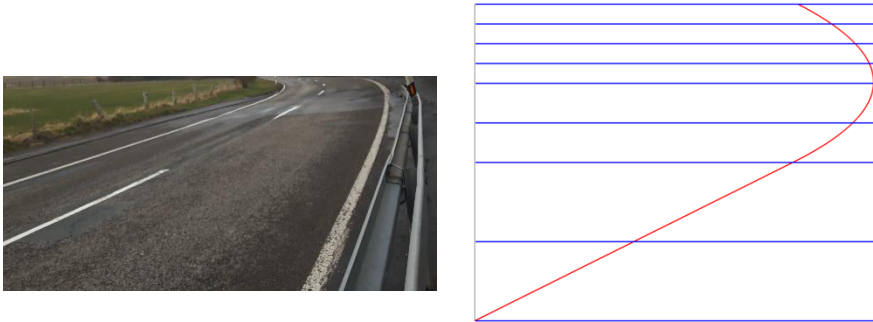


Figure 4

Lane detection: Original (cropped image) and interpolated function.

Blue lines mark the subimages.

The algorithms were tested on a subset of the CoPeD data set [21]. In the following, we give the relative path to the used sequences inside the folder that can be downloaded from the CoPeD website (Link: see Chapter 1):

- Crosswalk with traffic (CW w/ traffic), first 243 frames:
CoPeD\2 Crossings\Pedestrian\Pedestrian_Crosswalk_Traffic.mp4
- Crosswalk without traffic (CW w/o traffic), first 195 frames:
CoPeD\2 Crossings\Pedestrian\Pedestrian_Crosswalk_NoTraffic.mp4
- Straight, 74 frames:
CoPeD\1 Lane Detection\Pedestrian\Others\Pedestrian_Straight_2.mp4
- Left, 489 frames:
CoPeD\1 Lane Detection\Pedestrian\Others\Pedestrian_Left_2.mp4
- Right, 585 frames:
CoPeD\1 Lane Detection\Pedestrian\Others\Pedestrian_Right_2.mp4

Tables 2, 3, and 4 show the evaluation for road segmentation (Table 2), crosswalk detection (Table 3), and lane detection (Table 4). In the first two cases, we considered nine frames less than stated above, in the third case 14 frames less. This is because we computed the union of the last ten frames for road segmentation and formed the mean of different values of the last 15 frames for lane detection.

For road segmentation, an over detection means that in addition to the road, important data belonging to the background according to Figure 1 (traffic lights and signs) close to the user, at least 20×20 pixels in size, were extracted. An under detection means that not enough of the road was extracted in order to detect the road's markings.

For crosswalk detection, the goal was to detect the crosswalk in all frames of the first two sequences and no crosswalk in the other sequences. In the case of lane detection, it was the other way around: For the first two sequences, the result

should be ‘No lane detected’ whereas for the latter three the respective course of the road should be determined.

Table 2

Evaluation Road Background Segmentation (NF: Number of Frames, CD: Correct Detection, OD: Over Detection, UD: Under Detection)

	CW w/ traffic	CW w/o traffic	Straight	Left	Right	Total
NF	234	186	65	480	576	1541
CD	234	186	65	478	576	1539
OD	0	0	0	2	0	2
UD	0	0	0	0	0	0
Hit rate	100.00%	100.00%	100.00%	99.59%	100.00%	99.87%

Table 3

Evaluation Crosswalk Detection (NF: Number of Frames, CD: Correct Detection, ND: Not Detected if crosswalk is present, FP: False Positive)

	CW w/ traffic	CW w/o traffic	Straight	Left	Right	Total
NF	234	186	65	480	576	1541
CD	226	181	65	472	576	1520
ND	8	5	0	0	0	13
FP	0	0	0	8	0	8
Hit rate	96.71%	97.37%	100.00%	98.36%	100.00%	98.64%

Table 4

Evaluation Lane Detection (NF: Number of Frames, CD: Correct Detection, WD: Wrong Detection)

	CW w/ traffic	CW w/o traffic	Straight	Left	Right	Total
NF	229	181	60	475	571	1516
CD	229	181	60	473	541	1484
WD	0	0	0	2	30	32
Hit rate	100.00%	100.00%	100.00%	99.58%	94.75%	97.89%

The overall hit rates were 99.87% for road segmentation, 98.64% for crosswalk, and 97.89% for lane detection. It would be useful to compare the evaluations with results from the underlying ADAS algorithms, but this is difficult to achieve because they were tested on other, not publicly available data. Choi et al. [9] report 96.2% correctly detected, present crosswalks and 0.66% false positives for a data set of 21864 frames from which 1053 contain a crosswalk. In our case, the according numbers were with 96.9% and 0.76% in a similar range. For lane detection, Lee [23] tested on 1200 frames, if the car departed from the road or not. The reached hit rate was 96.42%. Beucher et al. [5] presented their results on a

small number of single frames. Even though it is not possible in all cases to compare our hit rates with the ones of the underlying ADAS algorithms, we can state that our adapted algorithms performed well and reached suitable hit rates that are applicable for ASVI.

4 Future Work

In the future, we will implement the ADAS algorithms on which the presented algorithms are based in order to be able to compare performance in the fields of ADAS and ASVI. The evaluation will be done using the CoPeD data set [21], which contains comparable video sequences from the driver and pedestrian perspectives. Implementing the ADAS algorithms is a challenging task, because the algorithms are in general not described in detail in the related papers.

Another possibility to provide information about the course of the road to a visually impaired pedestrian, is to adapt the ADAS algorithm from [11] which detects arrows on the road setting out lane arrangements.

Improving the segmentation will also improve the crosswalk and lane detection results. Therefore, we will consider up-to-date road segmentation algorithms using deep learning (e.g. [25]).

Additionally, we will focus on the remaining paths in Figure 1 and work on ASVI solutions based on ADAS algorithms for obstacle, especially bike and vehicle, detection (e. g. [13], [22], [35]) as well as traffic light (e. g. [27]) and traffic sign detection (e. g. [36]). The adaptations for all considered use cases will be summarized in a concept for transferring ADAS algorithms to ASVI using methods from software engineering [30] and project management [26].

Conclusions

The content in this article is embedded in our project of developing a concept for the transfer of camera-based algorithms from ADAS to ASVI.

In previous work, we identified a total of seven use cases that have to be considered. This article focuses on the detection of markings on the road, namely crosswalk and lane detection. In order to reduce the computational cost, we first introduced a road background segmentation. Thereby, crosswalk and lane detection can be carried out on the road only. Algorithms for traffic sign and traffic light detection, to be developed in the future, will be applied to the background part of the images, whereas obstacles will be detected in the whole image.

The presented road background segmentation is based on Beucher *et al.*'s [4] use of morphological watersheds and in addition used thresholds for the mean gray

and saturation values of each catchment basin. Our model reached a hit rate of 99.87%.

The basis of our developed crosswalk detection is Choi et al.'s idea of using a horizontal mean filter [7]. We combined the horizontal with a vertical filter and defined a decision process. The recognition rate of this algorithm was 98.64%.

The suggested lane detection used the Edge Distribution Function (EDF). Instead of applying the EDF to the whole image as in Lee's ADAS work [18], we first divided the image into subimages of increasing sizes from bottom to top. Afterwards, interpolation of the angles and analysis of the according function returned the course of the road. Correct detections occurred in 97.89% of the examined frames.

With that, we can state that the presented ASVI algorithms, adapted from relevant ADAS methods, achieved overall good hit rates and thus are applicable for ASVI.

In our future work, we will first compare the performance of the presented ASVI algorithms with their underlying ADAS solutions and then focus on the remaining use cases.

References

- [1] M. R. Adame, J. Yu, K. Moeller: Mobility support system for elderly blind people with a smart walker and a tactile map. Proceedings of XIV Mediterranean Conference on Medical and Biological Engineering and Computing, pp. 608-613, April, 2016
- [2] J. M. Alvarez, A. Lopez, R. Baldrich: Illuminant-invariant model-based road segmentation. Proceedings of the IEEE Intelligent Vehicles Symposium, pp. 1175-1180, June, 2008
- [3] J. M. Alvarez, A. M. Lopez: Road detection based on illuminant invariance. IEEE Transactions on Intelligent Transportation Systems, Vol. 12, No. 1, pp. 184-193, March, 2011
- [4] J. M. Alvarez, T. Gevers, Y. LeCun, A. M. Lopez: Road scene segmentation from a single image. Proceedings of the 12th European Conference on Computer Vision, pp. 376-389, Springer, October, 2012
- [5] S. Beucher, M. Bilodeau, X. Yu: Road segmentation by watershed algorithms. Proceedings of the Pro-art vision group PROMETHEUS workshop, pp. 1-24, April, 1990
- [6] R. R. Bourne, S. R. Flaxman, T. Braithwaite, M. V. Cicinelli, A. Das, J. B. Jonas et al.: Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis. The Lancet Global Health, Vol. 5, No. 9, pp. 888-897, September, 2017

-
- [7] S. Caraiman, A. Morar, M. Owczarek, A. Burlacu, D. Rzeszotarski, N. Botezatu, P. Herghelegiu, F. Moldoveanu, P. Strumillo, A. Moldoveanu: Computer vision for the visually impaired: the sound of vision system. Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 1480-1489, October, 2017
- [8] R. Cheng, K. Wang, K. Yang, N. Long, W. Hu, H. Chen, J. Bai, D. Liu: Crosswalk navigation for people with visual impairments on a wearable device. Journal of Electronic Imaging, Vol. 26, No. 5, pp. 053025-1 – 053025-14, September/October, 2017
- [9] J. Choi, B. T. Ahn, I. S. Kweon: Crosswalk and traffic light detection via integral framework. Proceedings of the 19th IEEE Korea-Japan Joint Workshop on Frontiers of Computer Vision, pp. 309-312, January/February, 2013
- [10] P. S. Duckett, R. Pratt: The researched opinions on research: visually impaired people and visual impairment research. Disability & Society, Vol. 16, No. 6, pp. 815-835, October, 2001
- [11] Z. Fazekas, P. Gáspár: Computerized recognition of traffic signs setting out lane arrangements. Acta Polytechnica Hungarica, Vol. 12, No. 5, pp. 35-50, 2015
- [12] M. Foedisch, A. Takeuchi: Adaptive real-time road detection using neural networks. Proceedings of the 7th International IEEE Conference in Intelligent Transportation Systems, pp. 167-172, October, 2004
- [13] D. Greene, J. Liu, J. Reich, Y. Hirokawa, A. Shinagawa, H. Ito, T. Mikami: An efficient computational architecture for a collision early-warning system for vehicles, pedestrians, and bicyclists. IEEE Transactions on Intelligent Transportation Systems, Vol. 12, No. 4, pp. 942-953, 2011
- [14] J. Fritsch, T. Kuehnl, A. Geiger: A new performance measure and evaluation benchmark for road detection algorithms. Proceedings of International IEEE Conference on Intelligent Transportation Systems, pp. 1693-1700, October, 2013
- [15] A. Haselhoff, A. Kummert: On visual crosswalk detection for driver assistance systems. Proceedings of the IEEE Intelligent Vehicles Symposium, pp. 883-888, June, 2010
- [16] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, C. Igel: Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. Proceedings of International Joint Conference on Neural Networks, pp. 1-8, August, 2013
- [17] J. Jakob, E. Cochlovius, C. Reich: Kamerabasierte Assistenz für Blinde und Sehbehinderte. InformatikJournal 2016/17, pp. 3-10, 2016, Available: <https://tinyurl.com/informatikJournal2016>

- [18] J. Jakob, K. Kugele, J. Tick: Defining camera-based traffic scenarios and use cases for the visually impaired by means of expert interviews. Proceedings of the IEEE 14th International Scientific Conference on Informatics, pp. 128-133, November, 2017
- [19] J. Jakob, J. Tick: Concept for transfer of driver assistance algorithms for blind and visually impaired people. Proceedings of the IEEE 15th International Symposium on Applied Machine Intelligence and Informatics, pp. 241-246, January, 2017
- [20] J. Jakob, J. Tick: Traffic scenarios and vision use cases for the visually impaired. Acta Electrotechnica et Informatica, Vol. 18, No. 3, pp. 27-24, September, 2018
- [21] J. Jakob, J. Tick: CoPeD: Comparable Pedestrian Driver data set for traffic scenarios. Proceedings of the IEEE 18th International Symposium on Computational Intelligence and Informatics, pp. 87-92, November, 2018
- [22] G. Kertész, S. Szénási, Z. Vámosy: Mult-directional image projections with fixed resolution for object matching. Acta Polytechnica Hungarica, Vol. 15, No. 2, pp. 211-229, 2018
- [23] J. W. Lee: A machine vision system for lane departure detection. Computer Vision and Image Understanding, Vol. 86, No. 1, pp. 52-78, 2002
- [24] R. P. Loce, R. Bala, M. Trivedi: Computer vision and imaging in intelligent transportation systems. John Wiley & Sons Ltd, West Sussex, 2017
- [25] G. L. Oliveira, W. Burgard, T. Brox: Efficient deep models for monocular road segmentation. Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4885-4891, October, 2016
- [26] P. Rosenberger, J. Tick: Suitability of PMBOK 6th edition for agile developed IT projects. Proceedings of the 18th International Symposium on Computational Intelligence and Informatics, pp. 241-246, November, 2018
- [27] S. Saini, S. Nikhil, K. R. Konda, H. S. Bharadwaj, N. Ganeshan: An efficient vision-based traffic light detection and state recognition for autonomous vehicles. Proceedings of the IEEE Intelligent Vehicles Symposium, pp. 606-611, June, 2017
- [28] S. Se: Zebra-crossing detection for the partially sighted. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2211-2217, June, 2000
- [29] R. Shilkrot, J. Huber, R. Boldu, P. Maes, S. Nanayakkara: FingerReader: A finger-worn assistive augmentation. In Assistive Augmentation, pp. 151-175, Springer, Singapore, 2018
- [30] I. Sommerville: Software Engineering. Pearson, Boston, 2011

- [31] M. Song, D. Civco: Road extraction using SVM and image segmentation. *Photogrammetric Engineering & Remote Sensing*, Vol. 70, No. 12, pp. 1365-1371, December, 2004
- [32] J. J. Van Rheede, I. R. Wilson, L. Di Bon-Conyers, S. Croxford, R. E. MacLaren, S. L. Hicks: Smart Specs: Electronic vision enhancement in real-life scenarios. *Investigative Ophthalmology & Visual Science*, Vol. 57, No. 12, p. 5165, 2016
- [33] S. Wang, Y. Tiang: Detecting stairs and pedestrian crosswalks for the blind by RGBD camera. *Proceedings of IEEE International Conference on Bioinformatics and Biomedicine Workshops*, pp. 732-739, October, 2012
- [34] A. Witzel: The problem-centered interview. *Forum: Qualitative Social Research*, Vol. 1, No. 1, Art. 22, January, 2000
- [35] M. T. Yang, J. Y. Zheng: On-road collision warning based on multiple foe segmentation using a dashboard camera. *IEEE Transactions on Vehicular Technology*, Vol. 64, No. 11, pp. 4974-4984, 2015
- [36] Y. Yuan, Z. Xiong, Q. Wang: An incremental framework for video-based traffic sign detection, tracking, and recognition. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 18, No. 7, pp. 1918-1929, 2017
- [37] Y. Zhai, G. Cui, Q. Gu, L. Kong: Crosswalk detection based on MSER and ERANSAC. *Proceedings of the IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 2770-2775, September, 2015