Pedestrian Crosswalk Detection Using a Column and Row Structure Analysis in Assistance Systems for the Visually Impaired

Krešimir Romić, Irena Galić, Hrvoje Leventić, Marija Habijan

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, Kneza Trpimira 2b, HR-31000 Osijek, Croatia
kresimir.romic@ferit.hr, irena.galic@ferit.hr, hrvoje.leventic@ferit.hr, marija.habijan@ferit.hr

Abstract: Computer vision-based approaches have become more common in assistance systems for the blind and visually impaired where portable devices can be used to assist users in their free movement. The method for pedestrian crosswalk detection with the main goal to facilitate the crossing of the road is proposed in this paper. The proposed crosswalk detection method is based on analyzing the image column and row structure. The Performance of this kind of approach relies on the input image resolution and quality. Therefore, guidance for selecting the appropriate input image resolution for this kind of approach is given. This approach is tested on the realistic input data captured with a monocular camera and using portable devices for image processing.

Keywords: assistance systems for the visually impaired; crosswalk detection; column structure; morphology

1 Introduction

The World Health Organization claims that there are 285 million people with some kind of visual impairment and 39 million of them are totally blind [1]. Most of them still do not use some advanced technological assistance systems, therefore, a white cane is the most widely spread assistance tool [2]. Advances in technology facilitate the development of advanced automated systems for navigation and orientation of blind and visually impaired people. This applies to the movement in both known and unknown environments where people are faced with different problems and obstacles. There are many types of problems and obstacles in the free movement of the blind and visually impaired, but this paper will deal with the problem of independent road crossing on designated zebra-style pedestrian crosswalks. This approach has the main goal to recognize a pedestrian crosswalk in front of a person in order to facilitate the road crossing on zebra crosswalks.

The proposed approach uses image processing techniques to get useful information about the environment from input frames from video sequences taken with a camera. Camera-based assistance systems are a very common solution for this problem [3], [4], [5]. In this case, we propose the method for pedestrian crosswalk recognition from video frames. The image processing techniques used for this purpose include white balancing, edge detection, and morphological operations extended with a specially tailored vertical and horizontal analysis of binary frames.

An algorithm is performed using a simple monocular camera and a portable computer as a processing device. The final output is information about the presence of a crosswalk in front of a person. Additional information about the position of the crosswalk in the image is also provided. The way of informing a person about the crosswalk is out of the scope of this paper, but some preliminary tests were conducted by using stereo sound signals [6]. Similarly, Mascetti et al. in [7] proposed the sonification of guidance data during road crossing.

This paper is organized as follows: the second chapter explains the important preprocessing actions; the third chapter explains the proposed method in detail; the fourth chapter provides the experimental results with the obtained performance and accuracy and the last chapter draws up a conclusion and provides guidelines for further work. Figure 1 shows a brief overview of the main steps of the proposed method.

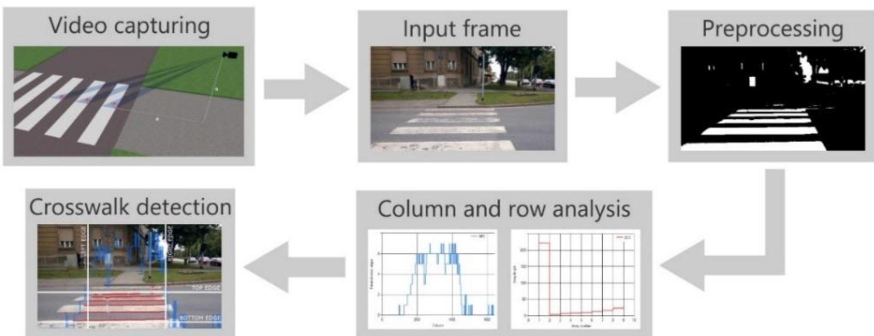


Figure 1

The method overview

1.1 Related Work

According to the survey [8], there are three groups of problems in individual movement of the blind: far, intermediate, and near distance tasks. Intermediate and near distance tasks like obstacle detection [9], space perception [10], and reading [11] are often solved using camera and image processing techniques.

Independent road crossing belongs to the group of intermediate distance tasks. Our previous work and similar approaches that deal with this problem based on image processing will be presented below.

When talking about zebra-style crosswalks, detection algorithms are often based on techniques used for detecting similar objects, e.g. staircases like in [4] and [12]. Our previous research on staircase detection proved that column-level analysis can be useful to localize free space and route the visually impaired people on the staircases. This research tries to implement and adopt similar principles to the problem of crosswalk detection. Furthermore, our previous research on crosswalk detection was more focused on potential crosswalk ROI extraction in preprocessing step in order to avoid whole image processing in higher resolutions, while this research presents the detailed column and row structure analysis method for crosswalk detection which yielded better results.

Similar to staircases, pedestrian crosswalks are characterized by rectangular shapes that appear periodically in an image making it suitable for using Hough transformation for detecting parallel lines as presented in [4]. The authors in [4] used RGBD camera which provides useful depth information; however, it is more expensive and impractical to use in motion. Detection of parallel vanishing lines is presented in [13] where a crosswalk slope is also approximated, but unlike the approach [4], the authors used a standard camera.

The use of a classic monocular camera is more frequent in real-time crosswalk detection systems. The mean shift segmentation and morphological processing are used to separate the crosswalk from the environment in [14] forming the robust and nonparametric system. The main disadvantages are lower detection rates due to larger shadows and very bright areas. Another approach [15] proposes a crosswalk detection algorithm based on bipolarity and projective invariant. This approach emphasizes the performance in various lighting conditions caused by sun, clouds or rain. In [16] and [17], figure-ground segmentation is used to detect crosswalks and the direction of the crosswalk is provided by finding a vanishing point. Figure-ground segmentation proved to be more robust than Hough transformation in cases of local deformations of straight lines which are often a part of crosswalk stripe edges. Cheng et al. [18] compared the conventional bipolarity method with a more novel adaptive extraction and consistency analysis (AECA) method on their own dataset in order to cope with inevitable aggravating capturing conditions. Adaptations of methods from driver assistance systems are also common solutions for crosswalk detection [19]. However, it can be concluded that the field of assistive technologies for the visually impaired is here neglected compared to vehicle and driver assistance systems, so this work certainly increases the awareness of this kind of problem.

Some authors recommend the development of algorithms tailored for execution on smartphones [3], [5] to take advantage of small devices with integrated cameras and additional help of GPS [20]. The mentioned smartphone-based approaches also investigate the usage of panoramic images in combination with satellite imagery. Authors in [21] present a software module called *ZebraRecognizer* with the main aim to remove projection distortion on acquired images to improve the crosswalk recognition accuracy.

In some recent scientific papers, machine learning techniques were used to solve crosswalk detection problems. Deep learning was successfully employed to detect crosswalks in [22] but the system is tailored for driver help and autonomous vehicles, so the camera perspective is different. The authors in [23] have used deep learning and they focus their research to detect crosswalks in different orientations. Authors also claimed the necessity for dataset extension to achieve better results with this approach. In [24], authors have dealt with crosswalk start and end-point detection with an accent on pedestrian traffic light recognition using convolutional neural networks. In mentioned work, testing was conducted only on images with crosswalks and lights. Even though the deep learning methods are becoming more common in this field, we have noticed that there is still a difference in approaches and methodology in such papers which, in combination with the lack of public datasets, makes them hardly comparable.

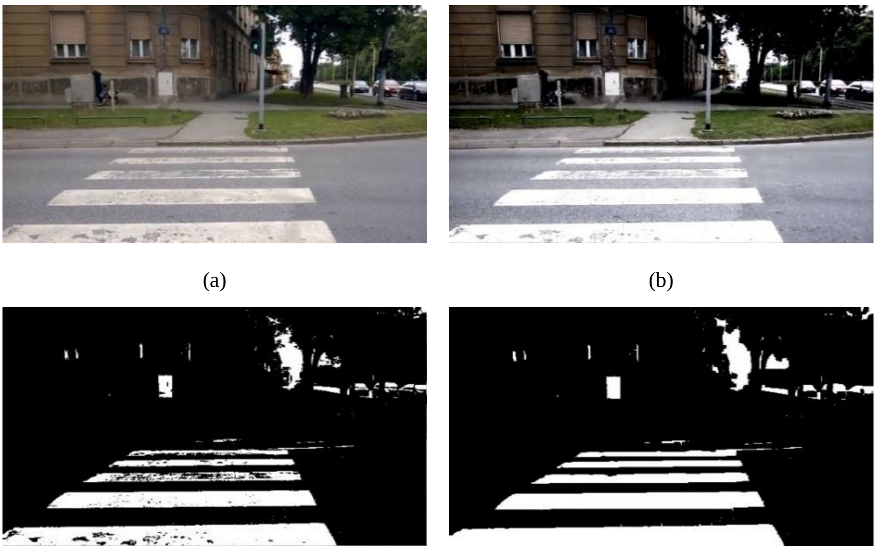
The shortcomings of the already proposed solutions can be summarized as follows: problems with shadows, obstacles and lighting conditions, problems with distortions caused by cameras, and the lack of unified test data especially for machine learning-based techniques.

Therefore, the goal of our method was in achieving good results without previous corrections regarding the projection distortions. Our approach actually uses the capturing perspective in order to detect characteristics of the crosswalk in particular camera views. This paper aims at a robust crosswalk detection method which will work in aggravating capturing conditions. Integrated crosswalk localization will be the additional benefit of the proposed method on the way to the final navigation system. The focus will be on the usage of medium and low-performance devices and cameras as the final goal is to develop a cheap and widely available system for the navigation of visually impaired persons.

2 Preprocessing

It is important to develop a method that will be able to work with frames captured from video sequences. Those frames can be considered as input data and a number of processed frames per second will show the performance of the method. The example of an input image is shown in Figure 2 a). The method is also geared to work on a video captured with a standard cheap monocular camera. Regarding capturing conditions and camera quality, we encountered several problems with input data. Those problems include different lighting conditions, unfocused and blurred frames due to camera movement. In order to solve the aforementioned problems, the preprocessing step enhances the visibility of the potential crosswalk and thereby prepares the image for further processing.

The initial preprocessing step starts with the white balancing of an input frame. Even though the human color perception of zebra crosswalk stripes indicates a white color, those stripes are often represented with slight gray and yellow shades in digital images depending on lighting conditions and camera specifications. To obtain clear white stripes of a potential crosswalk, white balancing is performed. For this purpose, we used a specialized Gray-world algorithm [25] to emphasize the white color as much as possible. This algorithm is chosen because of its simple implementation and low processing time. The effect of this process is presented in Figure 2 b).



(a) (b)

(c) (d)

Figure 2

Preprocessing: (a) Input image, (b) White-balancing, (c) Thresholding, (d) Closing

The next step is thresholding where near-white regions become white and all other colored regions become black. This step is necessary to emphasize the white regions of interest and thereby eliminate irrelevant image regions. Thresholding is performed on every color channel (red, green, and blue). For every pixel, all three channel intensities must be larger than the empirically obtained threshold value (220) and additionally, the difference between the two-channel intensities must be less than 20. Pixels that match the given criteria become white and all channels are set to value 255, otherwise to 0. The previous white color balancing allows us to use the fixed threshold values rather than adaptive methods thus reducing processing time. The final result of this step is a binary image as shown in Figure 2 c). The following equation shows thresholding conditions:

$$f^*(x,y) = \begin{cases} 255, & \begin{array}{c} if\ f_r(x,y)>220 \\ AND\ f_g(x,y)>220 \\ AND\ f_b(x,y)>220 \\ AND\ |f_r(x,y)-f_g(x,y)|<20 \\ AND\ |f_r(x,y)-f_b(x,y)|<20 \\ AND\ |f_g(x,y)-f_b(x,y)|<20 \end{array} \\ 0, & otherwise \end{cases} \tag{1}$$

where $f_r(x,y)$ represents the value of the red color channel of pixel with coordinates x, y. Similar to this, $f_g(x,y)$ represents the green channel and $f_b(x,y)$ the blue channel. Values that form the new binary image are represented with $f^*(x,y)$.

The final part of preprocessing is associated with possible imperfections on crosswalk stripes due to age and faded white paint on the road. First, the morphological closing operation is performed to fill narrow and small black regions with the white color [26]. The appropriate size of the structuring element for closing operation is chosen based on the resolution of the image which is explained in more detail in the experimental phase of this research (Chapter 4). An additional algorithm is tailored for finding and filling gray gaps on the road between the white color regions. Horizontal arrays of black pixels are compared in length with the surrounding horizontal white pixel arrays on the right and left. Every black array of pixels shorter than the surrounding white arrays is substituted with white pixels. This step successfully fills the most of black gaps between white horizontal regions which are characteristic of crosswalk stripes. The final binary image after the preprocessing step is shown in Figure 2 d).

3 The Method for Crosswalk Detection

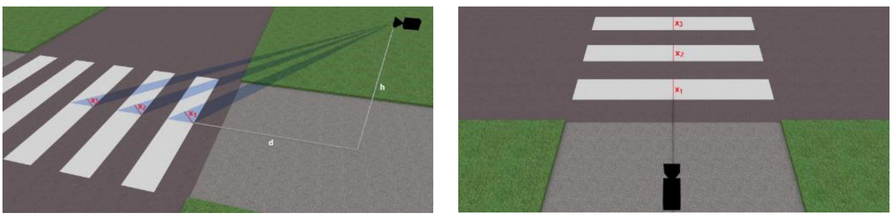
The main part of the proposed method for pedestrian crosswalk detection is explained in this chapter. The method is primarily based on the structure of image columns in the crosswalk region. The success of the method relies on the assumption that finding the specific characteristics of the column structure will allow us to detect which columns of the image belongs to the crosswalk region. The chapter is divided into three sections - column analysis, row analysis, and crosswalk localization.

3.1 Column Analysis

As opposed to the existing approaches which are often based on searching for nearly parallel horizontal lines [4], [5], [17], the proposed method is essentially based on a vertical analysis, i.e. a column analysis. The idea is to analyze every column of an input image and find features characteristic for a crosswalk region. If a camera is located on height h (approx. 1.5 m) and the crosswalk is in front of a camera on distance d (approx. 1-3 m), it is necessary to analyze how the white stripes are getting thinner with the increase of distance from a camera as shown in Figure 3 a). Further, in Figure 3 b), the camera perspective is shown with designated vertical red lines that represent white stripe width (x_1, x_2, x_3). It is obvious that those lines will gradually increase in length observing them from the top to bottom on captured video frames. It is experimentally obtained that the vertical white and black lines increase in length by 20% to 100% as shown in the following equation:

$$1.2 < k = \frac{x_n}{x_{n+1}} < 2.0 \tag{2}$$

where x_n is the length of n-th vertical array and k is the length increasing coefficient. It can be said that the vertical white and black lines increase in length in every particular column likewise members of geometric series, but with a certain aberration of k parameter. The estimated aberrations of k parameter shown in (2) are valid when the frame is captured from the distance 1-3 meters with a camera of 1.5 m height.



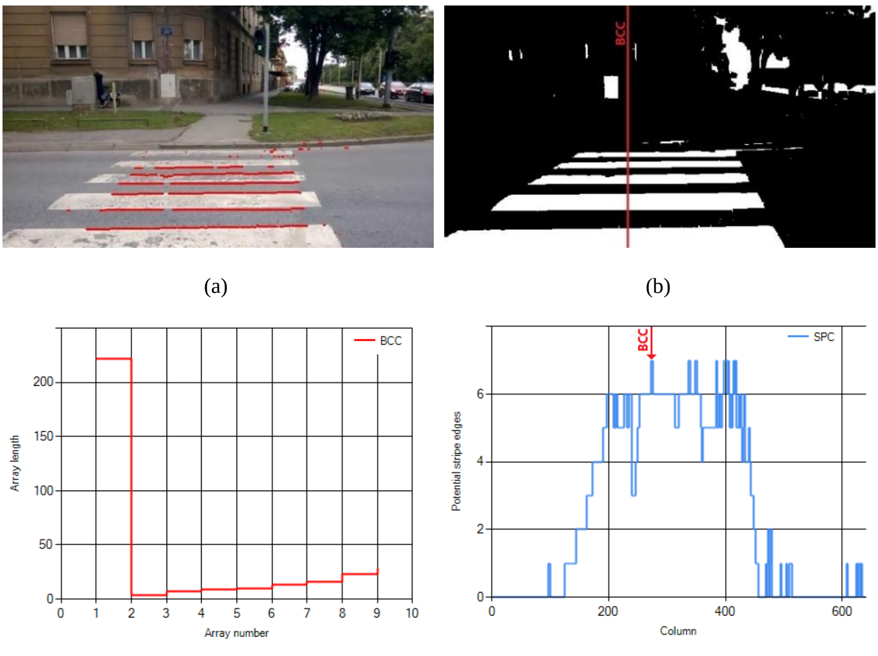
(a) (b)

Figure 3

Scene with camera and crosswalk: (a) Side perspective, (b) Camera perspective

An algorithm is developed to analyze every column of the binary image (Figure 4 b) obtained after preprocessing. The analysis is performed by measuring the length of the vertical white and black pixel arrays. The length of consecutive vertical pixel arrays of the same color (black or white) are compared and every increase that matches the required k parameter is counted as a potential stripe edge. Those potential stripe edges are marked with red dots on the input image illustrated in Figure 4 a). The vertical black pixel arrays are also tested because the space between white stripes acts similarly to stripes on the captured images.

When the column designated with a red line in Figure 4 b) is analyzed, the lengths of black and white vertical arrays have values as shown in Figure 4 c). To illustrate, this column is chosen because it is the column with the highest number of potential stripe edges and it is called the best column case (BCC). The graph in Figure 4 c) shows that arrays 2 to 9 gradually increase in length as requested in the mentioned condition (2) and 7 potential stripe edges are detected in that particular column. This process is repeated for every column and the graph of potential stripe edges per column (SPC) is generated (Figure 4 d). It is evident that the columns in the crosswalk region have much higher values. Supposing that a crosswalk consists of at least 2 stripes, columns with 4 or more potential stripe edges are counted. A higher number of such columns implies a higher possibility for finding the crosswalk in the image. Images with at least 10 columns with 4 or more potential stripe edges are forwarded to further analysis. Otherwise, images with less than 10 such columns are discarded and a decision about not finding a crosswalk is made. Once we have detected columns where crosswalk stripes are present we can use that information to find discontinuation in an array of crosswalk columns. This can help us to detect potential obstacles present on the crosswalk which is another benefit of this approach compared to related work.



(a) (b)

(c) (d)

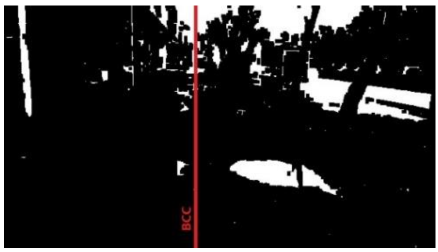
Figure 4

Column analysis (with crosswalk): (a) Detected potential crosswalk edges, (b) Analyzed binary image, (c) Array lengths in the best column case, (d) Potential stripe edges per column

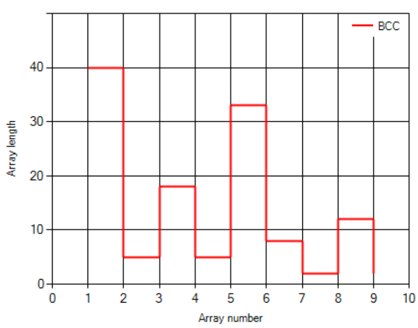
In comparison to the previous example, Figure 5 shows a detection process for an image example where a crosswalk is not present. Figure 5 a) shows the input image without the crosswalk where some points (red dots) were declared as potential stripe edges and those points are scattered on the entire image. The potential stripe edges are obtained by analyzing the binary image (Figure 5 b). If we observe a particular column with the highest number of potential stripe edges (red line in Figure 5 b), it is clear that the lengths of black and white vertical arrays do not have a constant increase in size characteristic for a crosswalk region. Those array lengths for the best column case are shown in Figure 5 c). The graph in Figure 5 d) shows that none of columns have more than 3 potential stripe edges. The overall number of potential stripe edges per column is much lower than in the previous case (Figure 4 d). An image like the one in Figure 5 a) is discarded at this moment considering that it does not have at least 10 columns with at least 4 potential stripe edges.

Putting the column analysis in the first place makes this method resistant to smaller obstacles that obstruct the view on a crosswalk. By having a larger range of parameter k values, the method is more resistant to different capturing angles, which are often problematic in similar approaches.

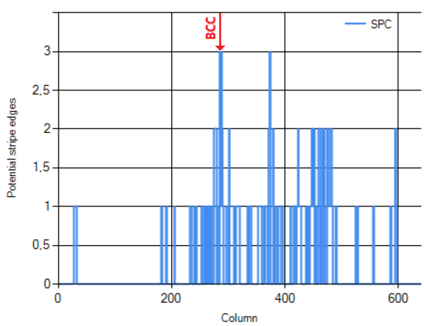
(a)



(b)



(c)



(d)

Figure 5

Column analysis (without crosswalk): (a) Detected potential crosswalk edges, (b) Analyzed binary image, (c) Array lengths in the best column case, (d) Potential stripe edges per column

3.2 Row Analysis

The last step of the method is the row analysis, i.e. the horizontal analysis which can be considered as an additional check. The row analysis is performed on a new image generated only from columns that passed the aforementioned column analysis with more than three potential crosswalk stripe edges. Generated images are narrower than the input image and are trimmed at the nearest and farthest potential stripe edge. In the trimming process, it is important to exclude isolated potential stripe edges that appear in distant positions from other potential edges because they unnecessarily increase the generated image size. The examples of generated images for the row analysis are shown in Figure 6.

(a)

(b)

Figure 6

Generated images for row analysis from input frame: (a) with crosswalk, (b) without crosswalk

The idea is to check whether the isolated columns are actually part of a crosswalk region or they have accidentally passed the previous step. The images generated from the input frames with a crosswalk (Figure 6 a) are characterized by wide rectangular shapes with lower rates of horizontal transitions from white to a black color and vice versa. In comparison, the images generated from the input frames without a crosswalk (Figure 6 b) are often narrower and have higher rates of horizontal transitions. Those images do not have distinctive horizontal stripes because they consist of columns from different parts of the input frame that accidentally passed the column analysis. In order to distinguish whether the generated images contain a crosswalk region or not, a special parameter called horizontal energy (HE) is calculated. The parameter is tailored to analyze row characteristics. It is calculated by the following equation:

$$HE = \frac{\sum_{i=0}^{h} l_{max}(i)}{\sum_{i=0}^{h} tr(i)} \tag{3}$$

where h is the image height, $l_{max}(i)$ is the length of the longest array of consecutive white or black pixels in row i, while $tr(i)$ is the number of transitions in row i. In this way, it can be decided whether the column analysis successfully detected a crosswalk region or it was a wrong assumption because HE will have higher values for images generated from the input frames with a crosswalk. On the other hand, images generated from the input frames without a crosswalk have lower HE. The threshold value for the final decision is obtained experimentally and is set to 2. Images with HE higher than 2 are marked as positive (i.e. contain a crosswalk) and other ones are discarded and marked as negative (i.e. do not contain a crosswalk).

3.3 Crosswalk Localization

Regarding assistance systems for the blind and visually impaired, it is important to provide simple information about a crosswalk position relative to a person [20]. If we observe data from a camera installed on a person, a vertical position of the crosswalk in the image is considered as irrelevant because it depends on a camera positioning angle and it is assumed that the crosswalk is on the same level as a person. On the other hand, it is important to emphasize the horizontal position of the crosswalk relative to a person. It can be very useful for a person to have real-time information whether the crosswalk is on the left, right, or in front.

In order to provide the crosswalk position on images, the previously shown layout and concentration of potential stripe edges (Figure 4 a) and Figure 4 d) are used. The combination of the required data is shown in Figure 7. When observing the columns from the left to the right, the first column with more than 3 potential stripe edges can be considered as the left edge of the crosswalk region. Similarly, the last column with more than 3 potential stripe edges is marked as the right edge of the crosswalk region. Those edges are marked with vertical white lines in Figure 7.

In images with the detected crosswalk, points that represent a potential stripe edge are concentrated in the crosswalk region, while a smaller number of points is isolated in other parts of the image. Those isolated points should be excluded and then the top and the bottom of the crosswalk can be approximately detected as the highest and the lowest potential stripe edge. The positions of those edges that determine the vertical position are marked with horizontal white lines in Figure 7. Having the left, right, top, and bottom edges, it is possible to outline the rectangle around the approximate position of the crosswalk. Other examples of localized crosswalks will be presented later in the chapter with the experimental results.

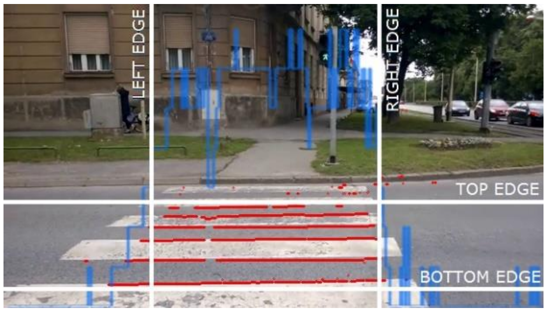


Figure 7
Crosswalk localization

4 Experimental Results

This chapter presents the experimental results obtained in realistic conditions. After initial experiment settings, the accuracy and performance of the proposed method are presented and compared to similar methods.

4.1 Experiment Settings

The proposed method is implemented in the C# programming language and the test environment is made for the sake of easier parameter monitoring and analyzing the method steps. The input video for processing can be captured directly from a camera or a video file. The resulting images of a particular method step can be analyzed for every frame of the input video along with a decision about detecting crosswalks and the crosswalk position.

The experiment was conducted by capturing video sequences in realistic conditions. The videos were captured in motion with a camera installed on a person at approximately the height of 1.5 meters (chest level). For this purpose, we used a standard monocular camera with 16:9 aspect ratio. Captured video sequences are prepared for testing in three different resolutions: 320×180, 640×360, and 1280×720. In contrast, to the RGB-D cameras used in a similar work [4], monocular cameras are cheaper and smaller, which makes them easier to use when they are installed on a person in motion. It is necessary to test the method on the frames from video sequences where people are approaching pedestrian crosswalks, but also from sequences without crosswalks. On video sequences that contain a crosswalk, it is important to capture a moment of approaching where a person is located on a distance of 1-3 meters from the crosswalk. A distance of 1-3 meters is a prerequisite for the proposed method to work properly.

Since this method is supposed to be used in assistance systems to help the blind in movement, two portable devices are chosen for the experiment. The first processing device is a laptop which brings an advantage in terms of higher computational power, but it is less portable due to its dimensions and weight. The second processing device is a mini PC with benefits in small dimensions, weight, and low power consumption, but with significantly less computational power.

4.2 Accuracy

When processing video sequences from files or directly from a camera, frames are processed one after another and real-time information about crosswalk presence is provided along with crosswalk approximate position. In order to obtain accuracy results, the representative frames were extracted from the video sequences for our database. There are 150 frames with a scene containing a crosswalk and 150 frames with a random scene not containing a crosswalk. It is crucial to test both groups of situations to avoid as much as possible false positive detections in scenes without a crosswalk. When collecting the input dataset, the videos captured in various aggravating conditions were chosen. This considers crosswalks with faded or soiled stripes on videos captured in different lighting conditions and from different angles. The input dataset does not include night scenes and the proposed method is primarily tailored to work with day scenes with natural light present. For night scenes, it would be necessary to use an additional lighting source, e.g. infrared lamp.

Testing is performed using images in three different resolutions and with three different structuring element sizes for closing operation as shown in Table 1. This experimental setup allows us to choose appropriate input resolution for this method and to choose the best closing parameter for a particular resolution. The results have shown that the resolution of 640×360 and closing structuring element size of 5×5 yield best accuracy rates. In that case, there are 98.7% correctly detected crosswalks and false-positive detections are present in only 1.3% of cases. These results are slightly better in comparison to our previous multiresolution approach [27].

Similar crosswalk detection accuracy is obtained on higher resolutions but with more false-positive detections due to unnecessary details visible in high-resolution images. Therefore, it can be concluded that it is unnecessary to use higher resolutions for this problem. On the other hand, it has been shown that lower resolution requires a smaller structuring element for closing and 76.67% of correct crosswalk detections can be obtained.

Table 1
Accuracy results based on image resolution and closing parameters

Resolution	320×180			640×360			1280×720		
Closing structuring element size	3×3	5×5	7×7	3×3	5×5	7×7	3×3	5×5	7×7
Detections (%) [with crosswalk]	76.7	52.7	22.7	94.7	98.7	93.3	94.7	97.3	97.3
Detections (%) [without crosswalk]	0	0	0	3.3	1.3	0.7	10	8.7	10

To our knowledge, there is no widely-used and available image collection for testing these kinds of methods, thus the direct comparison with the methods presented in the related work is not feasible. Authors mostly use their own databases for testing and there is a possibility that the results may be biased to some extent because the dataset is collected by the authors. Therefore, the proposed method is compared with our previous method on the same private dataset of 300 images. Furthermore, the proposed method is compared with the AECA (adaptive extraction and consistency analysis) method proposed in [18] on the dataset that authors provided online [28]. Comparison on our private database helps us to follow the progress of our own research in this field, while the comparison on the dataset provided by other authors helps us to test the robustness of our method on new images taken in different environments with different cameras. We found this comparison useful because mentioned work has the accent on crosswalk detection in aggravating conditions as is the case in this paper. Those conditions include pedestrian occlusion, low-contrast crosswalks, various illuminances, etc. Both datasets include images with and without crosswalks. We have measured the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for both comparisons and precision and recall metrics were calculated using the following equations (4).

$$Precision\ (\%) = \frac{TP}{TP+FP} * 100 \qquad Recall\ (\%) = \frac{TP}{TP+FN} * 100 \tag{4}$$

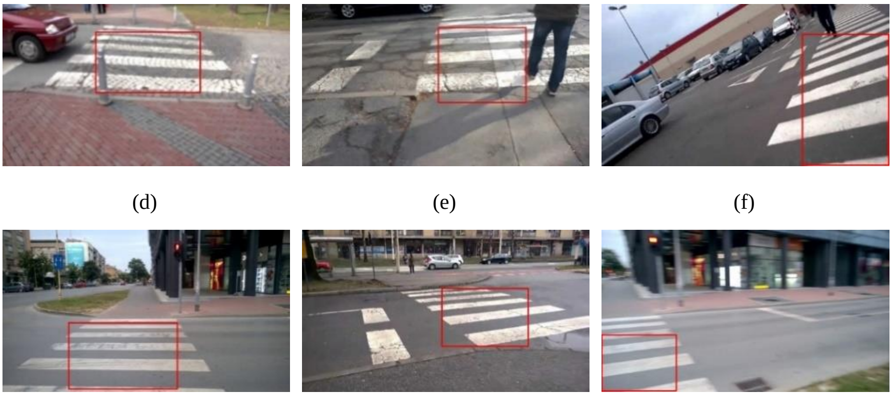
In the first comparison of our private dataset we have achieved slightly better results with precision and recall at 98.7% which is more than the previous 96.6% (precision) and 94.7% (recall). This method is computationally more demanding than the previous one but gives better accuracy rates which suggest that a certain trade-off between accuracy and processing speed is still unavoidable. The second comparison of the aforementioned public dataset implies that the proposed method is competitive with the AECA method. The proposed method gained a higher precision rate (94.8%) compared to the AECA method (84.6%). Similarly, the recall rate is also higher, 73.1% compared to 60.1%. It must be said that only overall results are compared and none of the subsets of data were tested individually. All accuracy rates of the proposed method compared to similar approaches are shown in Table 2.

Table 2
Accuracy results compared to similar methods

Dataset	Method	TP	TN	FP	FN	Recall (%)	Precision (%)
Dataset 1 (300 images)	Proposed method	148	148	2	2	98.7	98.7
	Previous multiresolution method	142	145	5	8	94.7	96.6
Dataset 2 (452 images)	Proposed method	239	112	13	88	73.1	94.8
	AECA algorithm	187	104	39	122	60.1	84.6

The examples of positive detections with localized crosswalks are shown in Figure 8. The test frames show that the method works very well in various situations. Figure 8 a) shows detection in usual daylight conditions, while Figure 8 b) shows detection on wet roads due to rain. Detection is also successful on faded crosswalk stripes (Figure 8 c) or on barely focused frames (Figure 8 d). Detection is possible even with the pedestrian presence on the crosswalk as shown in Figure 8 e). Figure 8 f) shows a situation when a camera is not in a completely horizontal position. Lower light conditions are present in the example in Figure 8 g). A situation when a person approaches the crosswalk at a certain angle is shown in Figure 8 h). The method proved to be robust to blurred frames as visible in Figure 8 i) due to thresholding and closing operation in the preprocessing step. For this reason, the deblurring procedure is not included in preprocessing in order to reduce processing time. Figure 8 i) also shows detection when the camera is not strictly directed to the center of the crosswalk and the pedestrian should be directed to the left side. Some test images show that missed detections are caused by shadows on a crosswalk surface, which makes binary images unusable for detection. Using the proposed method, false-positive detections are very rare and can occur only when shapes very similar to crosswalk stripes appear in an image.

(a) (b) (c)



(d) (e) (f)

(g) (h) (i)

Figure 8
Examples of positive detections (a)-(i)

4.3 Performance

According to [29], real-time implies that the response time must be enough to avoid a failure of the system. When talking about real-time video processing for this purpose, it is important to determine a minimal processing frame rate sufficient to beforehand inform a user about a crosswalk (i.e. before standing on the crosswalk). If we assume that the average walking speed is about 1.4 m/s [30] and the method detects a crosswalk from at least 1 meter distance, it is sufficient enough to process 2 frames per second because in that case, a user will only pass 0.7 meters while processing one frame. However, it is preferred not to take this information for granted because of a possible fault in particular frames due to capturing conditions in motion (e.g. unfocused and blurred frames). Furthermore, it would be good to make a final detection decision based on several consecutive frames, so it is preferable to process several frames per second.

The average processing speed obtained using the proposed method on input images in three different resolutions is given in Tab. 3. It is clear that the processing speed will also depend on the hardware specifications of processing devices, so two portable devices are chosen for the purpose of testing. The first device is a mid-range laptop and the second one is a mini PC with lower performance and power consumption. The detailed specifications of the used devices are also given in Tab. 3. Since we are aiming at developing a system that is dedicated only for assisting the visually impaired persons in their movement, we did not consider using smartphones as target devices. Even though it will be easy to deliver the system to potential users as a smartphone application, smartphones are also used for other everyday activities and the availability of processing resources could be occupied by other working applications on the smartphone or users can be distracted by other smartphone features. Another disadvantage is the battery life which can be very low due to demanding processing tasks. Therefore, we decided to test our method on a laptop, but also on a mini PC. This device uses 64-bit Intel Atom processor so it was convenient to use the same developed algorithm and libraries on both test devices. For further development of the fully usable assistance system, it will be necessary to adapt algorithms for embedded devices. However, for this stage of research we have found laptops and mini PC-s as good prototype platforms for our system.

Table 3
Processing speed on portable devices

		Laptop	Mini-PC
Specifications	Processor	Intel Core i5-3210M @ 2.50 GHz (2 cores)	Intel Atom Z3735F @ 1.33 GHz (4 cores)
	RAM	8 GB	2 GB
	Dimensions (W×D×H)	37.6×25.7×2.8 cm	10×3.8×1.5 cm
	Weight	2.5 kg	0.3 kg
Processing speed (per frame)	320×180 px	≈41 ms	≈158 ms
	640×360 px	≈150 ms	≈607 ms
	1280×720 px	≈634 ms	≈2579 ms

Using the 640×360 resolution frames, which yield the best accuracy results, the average processing speed is about 150 milliseconds on the laptop and about 607 milliseconds on the mini PC. When using the laptop, the information about the crosswalk presence is provided 6.67 times in a second, which can be considered as sufficiently fast to use such a system in reality. On the other hand, when using the mini PC with lower computational power, the information is provided 1.65 times in a second, which can be insufficient for some real situations. However, it is important to note that the size and weight of the mini PC is more suitable for use in motion. Since this paper puts the emphasis on accuracy, there is still enough space for algorithm optimization or parallel implementation, which will increase the processing speed. One of the solutions for improving the processing speed is to adapt the method to increase the accuracy on low-resolution images (e.g. 320×180) where a processing speed of nearly 24 fps is obtained.

Conclusion

The main contribution of this paper is a column-based approach for detecting crosswalks in video frames. This work brings some benefits to this research field by putting the accent on detecting the characteristic constitution of the columns in the crosswalk region. This concept allows robustness even when the part of the crosswalk is occluded. Additional contribution is proposed, horizontal energy parameter which is necessary for final decision making. This work has also brought the guidelines to choose the appropriate input resolution and preprocessing parameters for this kind of approach.

The proposed algorithm for crosswalk detection brings benefits to assistance systems for easier movement of the blind and visually impaired. The developed method for crosswalk detection is robust and yields high rates of correct detection. The method proved to be successful in aggravated conditions caused by lighting variations, faded crosswalk stripes, capturing angle, and camera shaking. Problems like variations in white color representation caused by shadows make space for the improvement of the method. Further research activities in this field will be focused on improving the method and adding additional features like detecting other pedestrians on the crosswalk.

The developed algorithm is tailored to perform on extensively used devices like laptops or mini PCs with a simple monocular camera. It is important to note that the proposed method is not dependent on camera type and the algorithm is tailored to work with images from different sources (e.g. web cameras, sports cameras, and camera glasses). Those facts make a potential assistance system cheaper and more discreet when installed on a blind person. The latter is very important not to additionally designate people with special needs in public.

The processing speed of the proposed algorithm is already sufficient for performing on mid-range laptops, but there is still space for optimization and improvements for performing on small devices like mini PCs. Although small devices will have increased computational power in the future, for now, it is necessary to make a compromise between the computational power and the size of a device. The final idea is to eventually develop a system that will integrate several types of aid for the movement of the blind. Therefore, the development and constant improvement of systems like the one presented in this paper is a necessity.

To sum up the contributions of this research it can be said that it brings better accuracy results proved on a private and public dataset. The presented column level approach brings easier crosswalk localization and potentially detection of obstacles on the crosswalk. Another benefit is accuracy and performance analysis on two different devices with several input image resolutions which is not common in related work and could be useful for further research.

References

[1] World Health Organization, "Global data on visual impairment." http://www.who.int/blindness/publications/globaldata/en/ (accessed Jul. 06, 2020)

[2] S. Shoval, I. Ulrich, and J. Borenstein, "NavBelt and the Guide-Cane [obstacle-avoidance systems for the blind and visually impaired]," *IEEE Robot. Autom. Mag.*, Vol. 10, No. 1, pp. 9-20, Mar. 2003, doi: 10.1109/MRA.2003.1191706

[3] V. N. Murali and J. M. Coughlan, "Smartphone-based crosswalk detection and localization for visually impaired pedestrians," in *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2013, pp. 1-7

[4] S. Wang, H. Pan, C. Zhang, and Y. Tian, "RGB-D image-based detection of stairs, pedestrian crosswalks and traffic signs," *J. Vis. Commun. Image Represent.*, Vol. 25, No. 2, pp. 263-272, Feb. 2014, doi: 10.1016/j.jvcir.2013.11.005

[5] V. Ivanchenko, J. Coughlan, and H. Shen, "Detecting and locating crosswalks using a camera phone," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008, CVPRW'08.*, 2008, pp. 1-8

[6] K. Romić, I. Galić, and T. Galba, "Technology assisting the blind - Routing on the staircases using wide-angle camera," in *2017 International Symposium ELMAR*, Sep. 2017, pp. 43-46, doi: 10.23919/ELMAR.2017.8124431

[7] S. Mascetti, L. Picinali, A. Gerino, D. Ahmetovic, and C. Bernareggi, "Sonification of guidance data during road crossing for people with visual impairments or blindness," *Int. J. Hum.-Comput. Stud.*, Vol. 85, pp. 16-26, Jan. 2016, doi: 10.1016/j.ijhcs.2015.08.003

[8] L. Hakobyan, J. Lumsden, D. O'Sullivan, and H. Bartlett, "Mobile assistive technologies for the visually impaired," *Surv. Ophthalmol.*, Vol. 58, No. 6, pp. 513-528, Nov. 2013, doi: 10.1016/j.survophthal.2012.10.004

[9] J. Zhang, C. W. Lip, S. K. Ong, and A. Nee, "A multiple sensor-based shoe-mounted user interface designed for navigation systems for the visually impaired," in *The 5th Annual ICST Wireless Internet Conference (WICON)*, Apr. 2010, pp. 1-8

[10] P. Strumillo, "Electronic Systems Aiding Spatial Orientation and Mobility of the Visually Impaired," in *Human – Computer Systems Interaction: Backgrounds and Applications 2*, Z. S. Hippe, J. L. Kulikowski, and T. Mroczek, Eds. Springer Berlin Heidelberg, 2012, pp. 373-386

[11] C. Yi, Y. Tian, and A. Arditi, "Portable Camera-Based Assistive Text and Product Label Reading From Hand-Held Objects for Blind Persons," *IEEEASME Trans. Mechatron.*, Vol. 19, No. 3, pp. 808-817, Jun. 2014, doi: 10.1109/TMECH.2013.2261083

[12] K. Romic, I. Galic, and T. Galba, "Technology assisting the blind - Video processing based staircase detection," in *ELMAR 2015 57th International Symposium*, Zadar, Sep. 2015, pp. 221-224, doi: 10.1109/ELMAR.2015.7334533

[13] S. Se, "Zebra-crossing detection for the partially sighted," in *IEEE Conference on Computer Vision and Pattern Recognition, 2000, Proceedings,* 2000, Vol. 2, pp. 211-217, doi: 10.1109/CVPR.2000.854787

[14] M. Radvanyi, B. Varga, and K. Karacs, "Advanced crosswalk detection for the Bionic Eyeglass," in *12th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA)*, 2010, pp. 1-5

[15] M. S. Uddin and T. Shioyama, "Robust zebra-crossing detection using bipolarity and projective invariant.," in *ISSPA*, 2005, pp. 571-574

[16] H. Shen, K.-Y. Chan, J. Coughlan, and J. Brabyn, "A mobile phone system to find crosswalks for visually impaired pedestrians," *Technol. Disabil.*, Vol. 20, No. 3, p. 217, 2008

[17] J. M. Coughlan and H. Shen, "A fast algorithm for finding crosswalks using figure-ground segmentation," presented at the 2nd Workshop on Applications of Computer Vision, in conjunction with ECCV, 2006

[18] R. Cheng *et al.*, "Crosswalk navigation for people with visual impairments on a wearable device," *J. Electron. Imaging*, Vol. 26, No. 5, Oct. 2017, doi: 10.1117/1.JEI.26.5.053025

[19] J. Jakob and T. József, "Camera-based On-Road Detections for the Visually Impaired," *Acta Polytech. Hung.*, Vol. 17, pp. 125-146, Jan. 2020, doi: 10.12700/APH.17.3.2020.3.7

[20] J. M. Coughlan and H. Shen, "Crosswatch: a System for Providing Guidance to Visually Impaired Travelers at Traffic Intersections," *J. Assist. Technol.*, Vol. 7, No. 2, Apr. 2013, doi: 10.1108/17549451311328808

[21] S. Mascetti, D. Ahmetovic, A. Gerino, and C. Bernareggi, "ZebraRecognizer: Pedestrian crossing recognition for people with visual impairment or blindness," *Pattern Recognit.*, Vol. 60, pp. 405-419, Dec. 2016, doi: 10.1016/j.patcog.2016.05.002

[22] V. Tümen and B. Ergen, "Intersections and crosswalk detection using deep learning and image processing techniques," *Phys. Stat. Mech. Its Appl.*, Vol. 543, Apr. 2020, doi: 10.1016/j.physa.2019.123510

[23] M. Poggi, L. Nanni, and S. Mattoccia, "Crosswalk Recognition Through Point-Cloud Processing and Deep-Learning Suited to a Wearable Mobility Aid for the Visually Impaired," in *New Trends in Image Analysis and Processing -- ICIAP 2015 Workshops*, Sep. 2015, pp. 282-289, doi: 10.1007/978-3-319-23222-5_35

[24] S. Yu, H. Lee, and J. Kim, "LYTNet: A Convolutional Neural Network for Real-Time Pedestrian Traffic Lights and Zebra Crossing Recognition for the Visually Impaired," in *Computer Analysis of Images and Patterns*, Cham, 2019, pp. 259-270, doi: 10.1007/978-3-030-29888-3_21

[25] A. Gijsenij and T. Gevers, "Color Constancy Using Natural Image Statistics and Scene Semantics," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 33, No. 4, pp. 687-698, Apr. 2011, doi: 10.1109/TPAMI.2010.93

[26] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson Education, 2011

[27] K. Romić, I. Galić, H. Leventic, and K. Nenadić, "Real-time Multiresolution Crosswalk Detection with Walk Light Recognition for the Blind," *Adv. Electr. Comput. Eng.*, Vol. 18, No. 1, pp. 11-20, Feb. 2018, doi: 10.4316/AECE.2018.01002

[28] Ruiqi Cheng, Zhejiang University, Hangzhou, China, "Pedestrian Crosswalks Recognition (PCR) Public Database." Zhejiang University, Hangzhou, China, Accessed: Mar. 20, 2020 [Online] Available: http://www.wangkaiwei.org/project.html

[29] Information Resources Management Association, *Image Processing: Concepts, Methodologies, Tools, and Applications*, 1 edition. Hershey, PA: IGI Global, 2013

[30] R. C. Browning, E. A. Baker, J. A. Herron, and R. Kram, "Effects of obesity and sex on the energetic cost and preferred speed of walking," *J. Appl. Physiol.*, Vol. 100, No. 2, pp. 390-398, Feb. 2006, doi: 10.1152/japplphysiol.00767.2005