

Symbol Detection in Low-Resolution Images using a Novel Method

Kuang-Chin Tai and Chien-Hsu Chen

Dept. of Industrial Design, National Cheng Kung University, Tainan, Taiwan
gqdai7@gmail.com; chenhsu@mail.ncku.edu.tw

Abstract

The ability to detect taxis, pedestrian walk symbols, zebra crossings, and straight line tactile guide paths and junctions is essential for blind people in any city. This work applies a novel method that detects automatically these five targets via their colors, geometric elements and structural features. To reduce processing time, this method is tailored to low-resolution images that are roughly 80 x 60 pixels. Additionally, another novel method based on the Hough Transform can recognize straight lines, an important geometric element, in low-resolution images. The effectiveness and efficiency of the proposed method are validated experimentally.

Keywords: *blind, navigation, assistive technology, object recognition, computer vision*

1. Introduction

Although sighted people can read signs and symbols to navigate urban areas, the blind cannot. Recognizing some of the many signs and symbols in cities are extremely important to the safety of the blind, such as for taxis, pedestrian walk symbols, zebra crossings, and straight line tactile guide paths and junctions. Detecting the symbols associated with these tasks is more than challenging for most blind people.

In recent years, several vision substitution devices have been developed [1]. After receiving a user's command, these devices provide specific services to individuals who have some sensory loss. Via an algorithm, these devices transform automatically data from a camera into a decision. This work applies a novel method to detect yellow taxis, pedestrian walk symbols consisting of a green walk icon and a yellow countdown timer, zebra crossings characterized by a regular pattern of white parallel stripes, and straight line tactile guide paths and junctions composed of bricks in the shape of yellow squares. The proposed method can be applied to vision substitution devices.

2. Related Work

A few methods or systems have been developed to detect pedestrian walk symbols, zebra crossings and/or tactile guide paths. Most target one or two symbols and provide extended functions, such as calculating the vanishing point of the ground in an image. These methods are characterized by their identification accuracy and processing time for tasks.

To detect pedestrian walk symbols, shape information of the icons is adopted. Wang, *et al.*, [2] extracted the Histograms of Oriented Gradients features from the hand figure in the red icon. Alghamdi, *et al.*, [3] applied the Speeded-Up Robust Features algorithm to an image to detect the green walk icon of a man with his legs apart, and to detect the red do-not-walk image of a man standing still. Shioyama, *et al.*, [4] detected the similar icons. They derived the affine moment invariant to determine the features of the man shapes. Ivanchenko, *et al.*, [5] also detected the man icon for walk. They analyzed the icon for the presence of two legs

and a head. In these studies, the exact distance from the camera to the symbols was not reported.

The size of any object in an image decreases linearly as the distance between the object and camera increases. This problem was also noted by Ivanchenko, *et al.*, [5] for walk symbol detection. In practice, this distance generally depends on road width as pedestrian walk symbols are typically on either sides of a road. For example, on a 14-meter-wide road, a walk symbol with icon sized 20×20 cm on the other side of the road can be 18×18 pixels when taken by a camera with a 30° angle of view and a 640×480 resolution. As the distance between the camera and target increases, an icon's size decreases and some details of shapes are eliminated. An icon's information in images weakens when icons in this study were shot at a long distance.

To detect signage for zebra crossings or tactile guide paths, information related to outline of stripe in the images were adopted. Ahmetovic, *et al.*, [6] extracted straight-line segments lying along long the edges for each stripe of a zebra crossing with a Line Segment Detector algorithm. Similarly, to extract the straight-line segments for each stripe in zebra crossings, Shen, *et al.*, and Ivanchenko, *et al.*, [7, 8] used a local grouping process based on figure-ground segmentation, and Wang *et al.*, Shioyama, *et al.*, and Se [2, 4, 9] used the Hough Transform. Xu, *et al.*, [10] also applied the Hough Transform to extract straight-line segments lying along a long pair of long edges of a tactile guide path in images.

By utilizing outline information, the algorithms that detect zebra crossings have relatively high accuracy because zebra crossing lines have very high contrast with pavement; that is, they comprise simple shapes (white stripes on a dark ground) and are large (serial long stripes). These algorithms accurately identify outlines. However, one must apply an edge detector, such as a Sobel or Canny edge detector, to detect edge points before extracting the locations of straight-line edge segments from an image.

These algorithms that utilize shape or outline information require an image with sufficient resolution so the shapes of icons or outlines of stripes can be rendered well in images. But, with a high-resolution image, the time taken by computing processes required increases.

Instead of focusing on shape or outline information, two algorithms mainly use color information to detect pedestrian walk symbols and tactile guide paths. To detect pedestrian walk symbols, Aranda and Mares [11] applied a model-based approach based on color histograms to recognize a rectangular area comprised of black, yellow and either green or red. The camera's focal lengths in their experiment were 6–15 m. Karungaru, *et al.*, [12] detected tactile guide paths by identifying the path's color using a color threshold and a neural network process. Although these two algorithms, which used color information, may not require high-resolution images, they detected only one symbol.

To the best of our knowledge, no detection algorithm has been designed for all five symbols targeted in this work. Moreover, no detection targets to detect taxis or junctions for tactile guide paths. According to group discussions with blind individuals, identifying these two targets are significant problems. In addition to junctions for tactile guide paths, Edwards, *et al.*, [13] showed that the blind have difficulty identifying junctions using their feet and canes. Although improvements to tiles [13, 14] were suggested, the area blind people can access was limited by leg length and cane length. This work proposes a method to that detects taxis, pedestrian walk symbols, zebra crossings, and straight line tactile guide paths and junctions on signs. This method can use low-resolution images of 80×60 pixels to reduce processing time. A technique based on the Hough Transform is applied to extract straight lines, one geometric element, from low-resolution images. In this work, a preliminary software program based on this algorithm has been already implemented using C++. The effectiveness of the software was evaluated using images.

3. Proposed Detection Method

3.1. Overview

This method detects taxis, pedestrian walk symbols, zebra crossings, and straight line tactile guide paths and junctions by recognizing their colors, geometric elements, and structural features in images. The algorithm resembles other algorithms in that it uses color information. It differs in that it does not use shape and outline information. Instead, geometric elements and structural information are used. This enables the algorithm to detect these targets in low-resolution images. The green walk icon in low-resolution images represents a cluster of a small area containing pixels of similar colors, as do countdown timers. Thus, small areas are regarded as points of geometric elements. In low-resolution images, stripes of a zebra crossing appear as straight lines, a geometric element; and a taxi in low-resolution images is a large area containing pixels of similar colors, and is regarded as a plane, another geometric element.

Colors, geometric elements and the structures of symbols in images can be characterized as follows: 1) a taxi is a yellow plane; 2) a pedestrian walk symbol is comprised of two points, the upper is yellow and the other is green, which are close to each other; 3) a zebra crossing is a group of white parallel lines; 4) a straight line tactile guide paths is one yellow line; and 5) junctions for tactile guide paths are comprised of two yellow lines that form a cross point in an image. These descriptions summarize what the algorithm “sees.”

The detection method has two modules: a color learning module and symbol detection module. These two modules are fed preprocessed low-resolution images. Preprocessing acquires RGB images from a digital camera or database and then reduces image resolution to 80×60 pixels. With an aid operator, the color learning module learns the colors of symbols. The learned color values are then stored for the symbol detection module. The symbol detection module detects automatically symbols in images.

3.2. Color Learning Module

The color learning module, as its name signifies, learns the colors of objects that appear in images captured from an environment. To express the R, G and B primary color values of the interested symbols, the RGB color space based on the Cartesian coordinate system is employed. In the RGB color space, each color appears in its primary spectral components of red, green and blue. Ideally, all colors of a single-color object captured from an environment are located at a point in the RGB color space. As an object typically has uneven surfaces and lighting angles differ, the colors of an object are influenced and are located in a larger region in the color space. To select this region, a cube can be used to range the colors. A range cube in this work is defined as the intersection of R, G and B color ranges. Thus, constructing the range cube of an object's color means establishing the ranges of R, G and B values

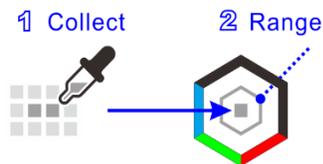


Figure 1. The Procedure to Establish Ranges of the R, G and B Values

The procedure applied to establish the ranges of R, G and B values has two stages (Figure 1).

(1) Manually collect the number of colors of the target object in images. In this experiment, 30 colors are collected.

(2) Range the R, G and B values based on collected colors, and store the range values. The equations to establish one primary color range (R, G or B) are as follows:

$$W = (q - p)/2 \times S$$

$$range = (q + p)/2 \pm W$$

where q and p are the highest and lowest value of the primary color values of the collected colors; S is an adjustment parameter, which is set manually to fit the range tightly to colors of interest in an RGB color space; and the width of W is half of the range. Table 1 lists the parameters and primary color range values for the colors of symbols in this experiment, and Figure 2 shows the location of range cubes in the RGB color space.

Table 1. Parameters and Primary Color Range Values for the Colors of Symbols in the Experiment

	S	R			G			B			
		p	q	Range	p	q	Range	p	q	Range	
Taxi	1.0	132	255	132 255	103	249	103 249	0	111	0 111	
Walk	Timer	1.1	190	255	187 258	157	255	152 260	0	123	-6 129
	Icon	1.1	31	144	25 150	179	255	175 259	162	255	157 260
Crossing	1.0	164	254	164 254	165	254	165 254	159	252	159 252	
Path	0.9	121	173	124 170	122	158	124 156	88	123	90 121	

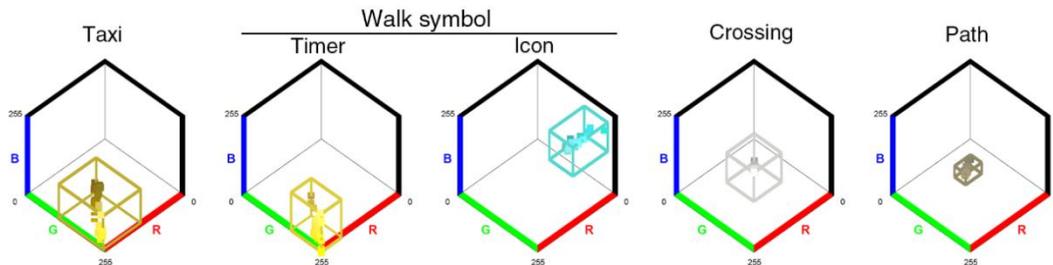


Figure 2. The RGB Color Spaces and Range Cubes Built in the Experiment. The Collected Colors are Located in the RGB Color Spaces with a Color According to their R, G and B Values

In Figure 2, the range cubes for taxis, countdown timer of pedestrian walk symbols and tactile guide path differ by location and size, even though all are “yellow” objects. Thus, the yellow of tactile guide paths is darker than the others two yellows. This may be caused by the texture of the bricks, which tend to retain dust. The range cube of taxis is larger than the others two yellow objects, likely because cars have curved surfaces and are glossy. Further, the “green” of the walk icon is not pure green in an image; it resembles cyan. This was also found by Kim, *et al.*, [15] when constructing a traffic light recognition system. Additionally, this work identified that the size of range cubes of countdown timers and walk icons are similar, and both objects are illuminated. Additionally, some “white” stripes of zebra crossing range from white to gray. The color learning module has the advantage of charactering range cubes without many trial-and-error attempts.

3.3. Symbol Detection Module

This module uses four procedures to detect the five garget symbols (taxis, pedestrian walk symbols, zebra crossings, and straight line tactile guide paths and junctions); each procedure detects one symbol, except the tactile guide path detection procedure detects straight lines and junctions (Figure 3). The detection procedures detect symbols via their colors, geometric elements and structural features that represent a symbol's appearance in an image to narrow the search space. The principles of color, geometric element and structural feature stages are as follows.

(1) The color feature stage converts RGB images into indexed color images with range cubes, which are previously constructed in the color learning module. All pixel colors in an RGB image are projected into the RGB color space, and any pixel colors projected in the range cubes are replaced with an indexed color, and all other pixels are set to white.

(2) The geometric element stage identifies planes, points and/or straight lines for regions or pixels of indexed colors. To detect a plane and point on geometric elements, a depth-first search algorithm is employed. Pixels that are 8-adjacent to each other are grouped. A group is a plane if its number of the pixels exceeds a plane threshold; otherwise, a group is a point if being smaller than a point threshold. In the experiment, the plane threshold is 1200 and point threshold is 16. To detect straight lines, a Hough Transform-based algorithm is applied. This algorithm extracts straight lines from low-resolution images.

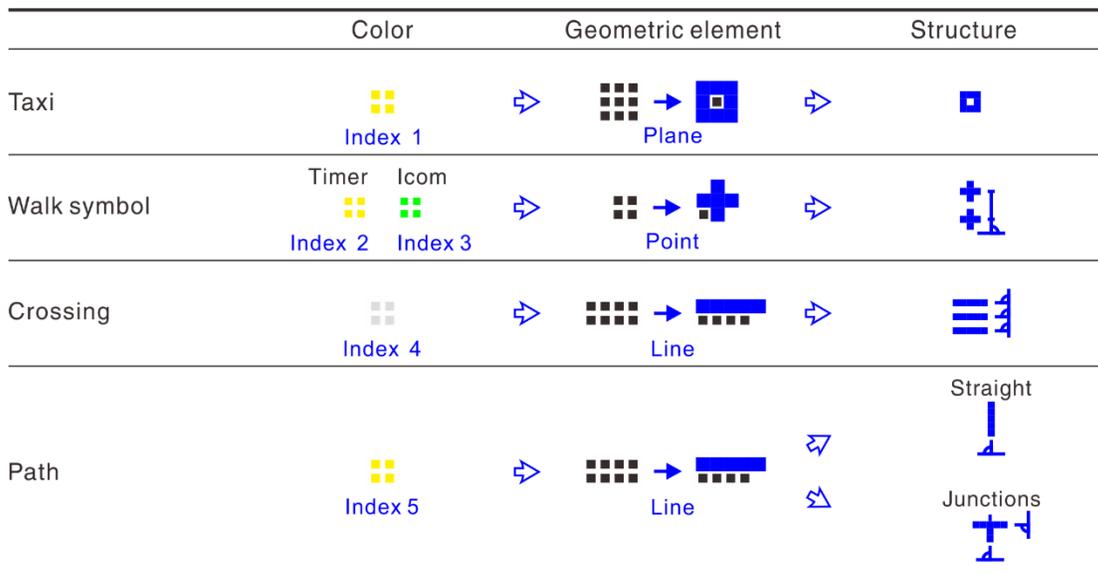


Figure 3. The Detection Procedures for Taxis, Pedestrian Walk Symbols, Zebra Crossings and Tactile Guide Paths. The Tactile Guide Path Detection Procedure Detects Straight Lines and Junctions for Tactile Guide Paths

(3) The structural feature stage measure and recognize the relationships between detected objects, the cross point of straight lines and/or the image frame.

To detect taxis, yellow is segmented from a low-resolution image using the taxi range cube. The plane elements are detected for yellow in the indexed color image. As an example in taxi sequence (Figure 4), a yellow plane is detected and is marked with a blue rectangle. The structural feature stage checks whether a yellow plane exists. If so, this procedure detects a taxi.

To detect pedestrian walk symbols, yellow and green are segmented using the countdown timer and walk icon range cubes. As an example in walk symbol sequence (Figure 4), two yellow points and one green point are detected, and are marked with a small blue cross. The structural feature stage recognizes a structure, a yellow point above to a green point; these two points are close to each other and are at a nearly vertical angle in the image. The distance limit threshold between two points is 8.5 pixels and the angle limit threshold is 30° to vertical in the image.

To detect zebra crossings, white is segmented using the zebra crossing range cube. The white lines are then extracted from the image. The structural feature stage recognizes groups of lines that are nearly parallel. In the experiment, a group has at least 3 lines, and the angles of the lines that difference from the line at top of the group is less than 1° .

For tactile guide path detection, yellow is segmented using the tactile guide path range cubes. Then, yellow lines are extracted from the image. The structural feature stage determines the number of yellow lines. If only 1 line exists, the stage measures the line's angle and determines whether it is a vertical line used to identify a straight line tactile guide path. If 2 lines exist, the stage calculates the location of the cross point of the two lines and determines whether lines exist around the cross point. As an example in junctions sequence (Figure 4), when a line that is not in the upper side of the cross point, the structure of the existing lines is recognized as a left/right junction.

	Low resolution	Color	Geometric	Structure	Result
Taxi					
Walk symbol					
Crossing					
Straight for path					
Junctions for path (left/right)					
	Junctions for path (Left fork)		Junctions for path (Right fork)		

Figure 4. Examples of the Four procedures that Illustrate Process Results using Colors, Geometric Elements and Structural Features

4. Extract Straight Lines in Low-resolution Images

Generally, using the Hough Transform to detect straight lines in images uses two stages. The first stage converts a binary image into a 2D accumulator array via the Hough Transform. The second stage detects local maxima in the accumulator array. In the first stage, each pixel of objects in an image is converted into a sinusoidal curve in the accumulator array. If the image has a straight line, a peak (local maximum) will be generated in the accumulator array.

The Hough Transform algorithm has been widely used [2, 4, 9, 10] to detect straight lines to identify zebra crossings and tactile guide paths in images. Edge detection was utilized before applying the Hough Transform to images to detect straight lines for edges; however, the proposed algorithm does not utilize edge detection. As shown in junctions sequence (Figure 4), the tactile guide paths in the index color image present as broken stripes. These broken stripes may not result in ideal straight edges, especially with low-resolution images. Hence, index color images are applied with the Hough Transform without utilizing edge detection in the proposed algorithm, while stripes in images are so wide and broken that local maxima generated from these stripes do not have significant high values.

One way to detect local maxima in the accumulator array involves using a threshold. Local maxima whose values are greater than the threshold correspond to straight lines. A common threshold is defined by some fixed percentage of the global maximum value. But, this threshold dose not identify whether the local maxima is increased from straight lines or not. In the proposed algorithm, an adaptive threshold is determined by the equation $\text{Sum}/\text{Count} \times F$, where Count is the total number of cells whose value is not zero; Sum is the sum of values of cells in the accumulator array; and F is a fixed parameter that is set experimentally. For a 1D accumulator array example (Figure 5), the threshold is 3.5 and the maximum cell's value is 5, such that this cell is a local maximum cell corresponding to a straight line.

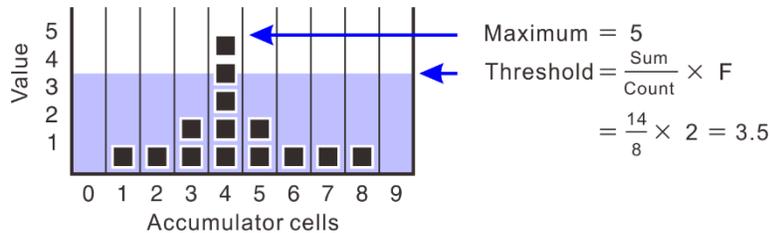


Figure 5. An example of Determining a Threshold to Identify a Local Maximum on a 1D Accumulator Array

Table 2. An Experiment to Assess the Method's Accuracy in Terms of Differentiating between a Line as a Geometric Element and a Point and a Plane in Low-resolution Images. The Column "VS." is Result of Comparing the Maximum with the Threshold

Image	Accumulator	Sum	Count	F	Maximum	Threshold	VS.
		380,340	10,470	2	64	72.65	-
		11,340	1,787	2	10	12.69	-
		113,580	10,038	2	79	22.63	+

An experiment was conducted to assess the accuracy of this method in terms of distinguishing lines from points and planes in low-resolution images (80×60 pixels) (Table 2). The F parameter is set to 2. As a comparison of the maximum value with the threshold for each geometric element in an image, only the maximum that is generated from a line exceeds the threshold, such that an image with a line is identified as different from the other two images.

5. Experiments and Discussion

The experiment used images to gauge the effectiveness and efficiency of the proposed algorithm. The equipment was an Intel Core i5 2.5-GHz PC. Images were collected with a Sony Cyber-shot DSC-P10C camera with an angle of view set at 30° . The images were taken within one week, mostly sunny days, in a two-hour period from 4:00–6:00 p.m. to homogenize lighting. Distances from the camera to target symbols were about 4 m, except the pedestrian walk symbols were shot at between 12–16 m. Target symbols were captured in different places except the straight line tactile guide paths and the junctions since the colors of tactile guide paths differed markedly from location to location. For junctions, only the left/right, left fork and right fork tactile guide paths were tested, and were found at one location.

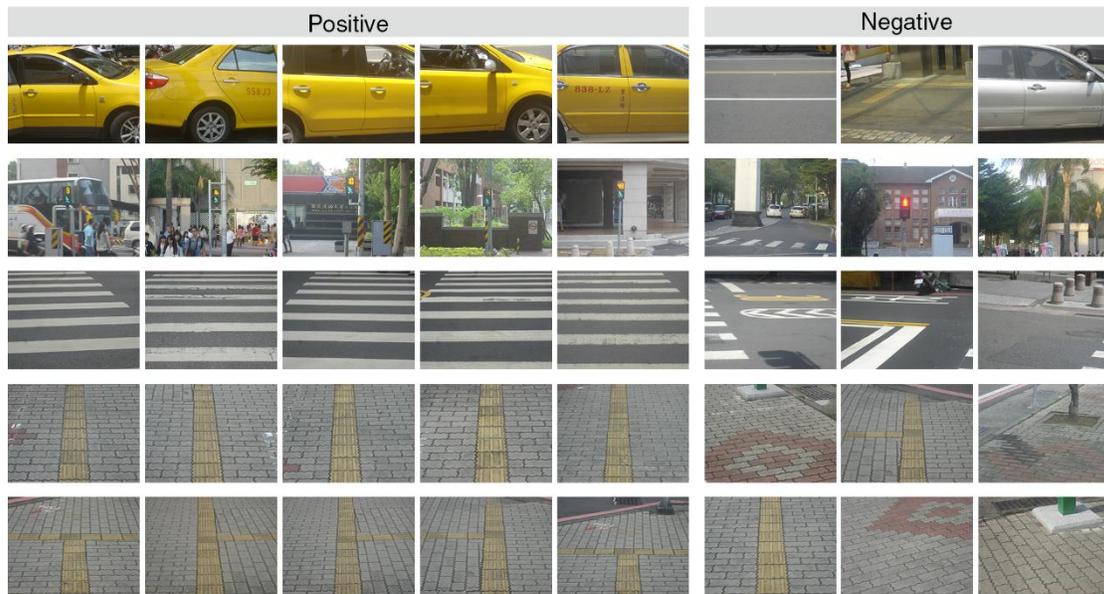


Figure 6. Examples of the Five Datasets: Taxis (row 1), Pedestrian Walk Symbols (Row 2), Zebra Crossings (Row 3), and Straight Line Tactile Guide Paths (Row 4) and Junctions (Row 5)

Five datasets of images were generated for the five target symbols (Figure 6). Each dataset had positive and negative images. The positive images contained target symbols, and negative images did not. Figure 7 shows the conditions for acquiring positive and negative images. In positive conditions, the camera was directed at the symbols; in negative conditions, the camera was aimed at places near the symbols. The purpose of the positive and negative test images is to approximate a use case. By using vision substitution devices, blind users may or may not capture a specific symbol they would like to identify. Each dataset had 10 images for the color learning module to learn the colors of the target symbols before testing the symbol

detection module. The four symbol detection procedures in the symbol detection module were tested with the five symbol datasets (The tactile guide path detection procedure was tested with two datasets: a straight line dataset and a junction dataset for a tactile guide path).

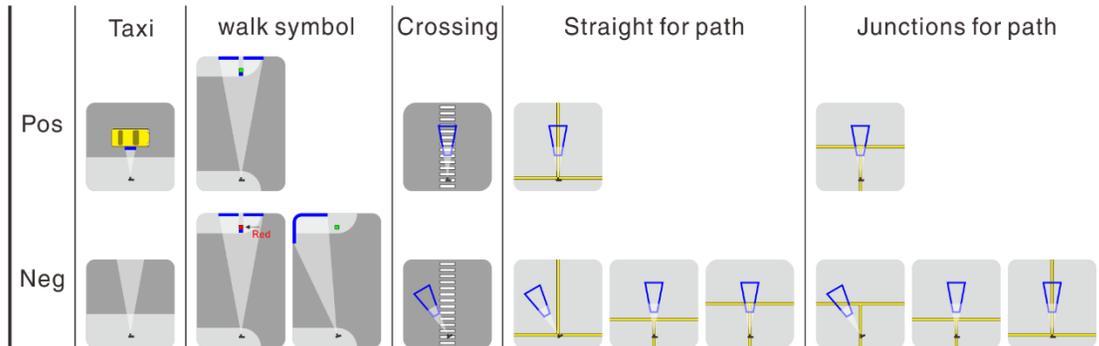


Figure 7. Plan View of the Image Collection Conditions in this Experiment. This Illustrates the Camera's Locations and Angles for Positive and Negative Images. Blue Lines Frame the Regions of the Scenes Captured

Table 3. Test Results with the Five Symbol Datasets

	Positive				Negative			
	Num	Corr	Accy	Time	Num	Corr	Accy	Time
Taxi	93	93	100.00	7.28	64	64	100.00	1.40
Walk symbol	263	257	97.72	2.09	125	119	96.57	1.91
Crossing	259	259	100.00	37.54	81	80	98.77	35.12
Straight for path	41	41	100.00	24.98	35	35	100.00	19.66
Junctions for path	90	90	100.00	30.21	60	60	100.00	21.53
Average			99.54	20.42			98.79	15.93

Table 3 lists test results with the five symbol datasets. The columns in Table 3 are headed 'Num' for number of image samples, 'Corr' for number of image samples correctly identified, 'Accy' for percentage of image samples correctly identified, and 'Time' for average processing time (millisecond) for each image.

Test results (Table 3) show that the proposed algorithm achieves a 99.54% true positive rate for positive image samples and 98.79% true negative rate for negative image samples. Average processing time for positive image samples was about 20.42 milliseconds and that for negative image samples was about 15.93 milliseconds. A few false results existed for positive image samples of the pedestrian walk symbol. In most cases (Figure 8(a)), as getting colors in a source image, the cell array completely misses the shape of the countdown timer. Otherwise, a few false results existed for negative image samples of the pedestrian walk symbol. In most cases (Figure 8(b)), parts of objects which randomly possess the "countdown timer yellow" and "walk icon green" form a new whole and thus make falsely identification. One false result was generated for identifying negative image samples for a zebra crossing (Figure 8(c)). The parts of the road mark captured in the image forms a group of parallel lines. Overall, experimental results are promising and demonstrate the effectiveness and efficiency of the proposed algorithm in detecting target symbols.

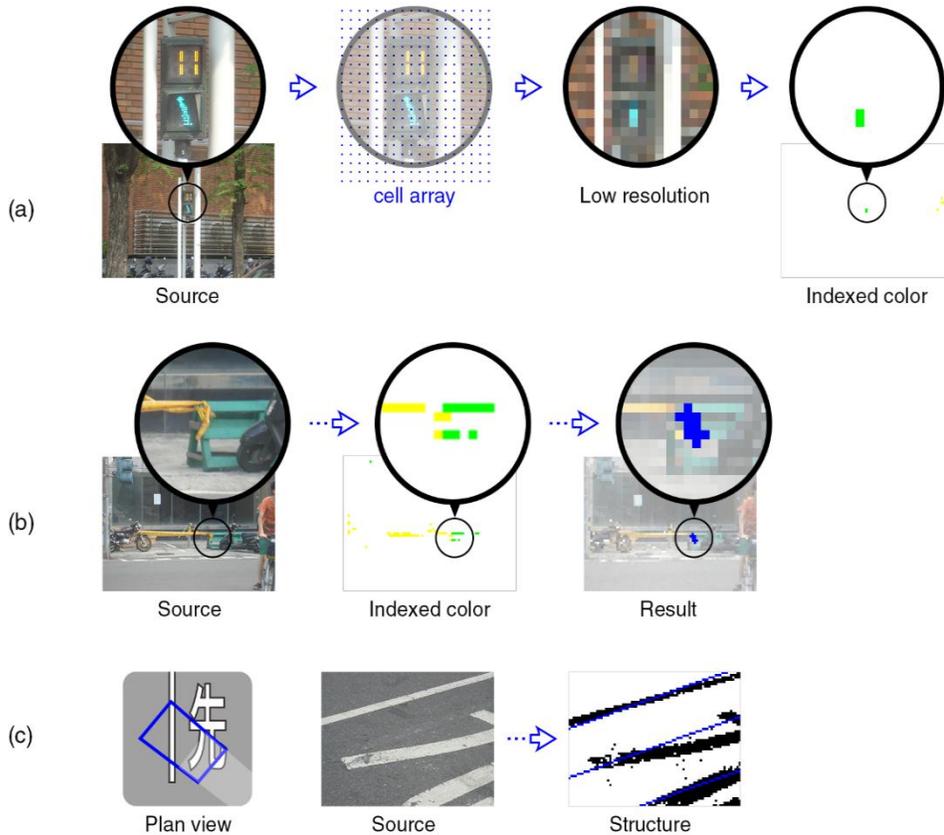


Figure 8. Examples of False Results in the Experiment

6. Conclusions and Future Work

This work applies a novel method to detect taxis, pedestrian walk symbols, zebra crossings, and straight line tactile guide paths and junctions by their colors, geometric elements and structural features on low-resolution images. This method achieves a high accuracy rate and speed. The pedestrian walk symbols can be detected at distances of 12–16 m from the camera, yielding a 97.72% true positive rate for positive image samples and a 96.57% true negative rate for negative image samples. A zebra crossing can be detected in an average processing time of roughly 37.54 milliseconds for positive image samples and approximately 35.12 milliseconds for negative image samples. Future research will focus on enhancing the effectiveness of the proposed algorithm in detecting these target symbols under different lighting conditions.

Acknowledgements

The authors would like to thank the Taiwan Foundation for the Blind for their assistance with recruitment of participants.

References

- [1]. R. Jafri, S. A. Ali, H. R. Arabnia and S. Fatima, “Computer vision-based object recognition for the visually impaired in an indoors environment: a survey”, *The Visual Computer*, (2013).

- [2]. S. Wang, H. Pan, C. Zhang and Y. Tian, "RGB-D image-based detection of stairs, pedestrian crosswalks and traffic signs", *Journal of Visual Communication and Image Representation*, vol. 25, no. 2, (2014), pp. 263–272.
- [3]. S. Alghamdi, R. Van Schyndel and I. Khalil, "Safe trajectory estimation at a pedestrian crossing to assist visually impaired people", *Proceedings of Medicine and Biology Society*, (2012), pp. 5114–5117, San Diego, USA.
- [4]. T. Shioyama, H. Wu, Y. Nishibe, N. Nakamura and S. Kitawaki, "Image analysis of crosswalk", *Proceedings of Image Analysis and Processing*, (2001), pp. 168–173, Palermo, Italy.
- [5]. V. Ivanchenko, J. Coughlan and H. Shen, "Real-time walk light detection with a mobile phone", *Lecture Notes in Computer Science*, vol. 6180, no. 2, (2010), pp. 229–234.
- [6]. D. Ahmetovic, C. Bernareggi and S. Mascetti, "ZebraLocalizer: Identification and localization of pedestrian crossings", *Proceedings of Human Computer Interaction with Mobile Devices and Services*, (2011), pp. 275–284, Stockholm, Sweden.
- [7]. H. Shen, K.-Y. Chan, J. Coughlan and J. Brabyn, "A mobile phone system to find crosswalks for visually impaired pedestrians", *Technology and Disability*, Vol. 20, no. 3, (2008), pp. 217–224.
- [8]. V. Ivanchenko, J. Coughlan and H. Shen, "Detecting and locating crosswalks using a camera phone", *Workshop of Computer Vision and Pattern Recognition*, (2008), pp. 1–8, Anchorage, USA.
- [9]. S. Se, "Zebra-crossing detection for the partially sighted", *Proceedings of Computer Vision and Pattern Recognition*, vol. 2, (2000), pp. 211–217, Hilton Head Island, USA.
- [10]. J. Xu, X. Wang and Z. Fang, "Research and implementation of blind sidewalk detection in portable ETA system", *Proceedings of Information Technology and Applications*, vol. 2, (2010), pp. 431–434, Kunming, China.
- [11]. J. Aranda and P. Mares, "Visual system to help blind people to cross the street", *Lecture Notes in Computer Science*, vol. 3118, (2004), pp. 454–461.
- [12]. S. Karungaru, K. Terada and M. Fukumi, "Improving mobility for blind persons using video sunglasses", *Workshop of Frontiers of Computer Vision*, (2011), pp. 1–5, Ulsan, South Korea.
- [13]. M. C. Edwards, "Guidance path at station square, Milton Keynes", *Traffic Engineering and Control*, vol. 34, no. 11, (1993), pp. 545–550.
- [14]. A. Courtney and H.M. Chow, "A study of tile design for tactile guide pathways", *International Journal of Industrial Ergonomics*, vol. 25, no. 6, (2000), pp. 693–698.
- [15]. Y. K. Kim, K. W. Kim and X. Yang, "Real time traffic light recognition system for color vision deficiencies", In *Proceedings of Mechatronics and Automation*, (2007), pp. 76–81, Harbin, China.

Authors



Kuang-Chin Tai. He got his M.S. at Dept. of Visual Communication Design of Kun Shan Univ. at 2005. Currently, he is a Ph.D. candidate at Dept. of Industrial Design of National Cheng Kung Univ. His research interest includes computer vision and assistive technology.



Chien-Hsu Chen, He got his Ph. D. at Dept. of Industrial Engineering of Texas Univ. at Arlington. He is a professor at Dept. of Industrial Design of National Cheng Kung Univ. since 1998. His research interest includes ergonomics and interaction design, augmented reality and usability design.

