Technology and Disability 20 (2008) 1–8 1
IOS Press

A mobile phone system to find crosswalks for visually impaired pedestrians

Huiying Shen[a], Kee-Yip Chan[b], James Coughlan[a,*] and John Brabyn[a]
[a]*The Smith-Kettlewell Eye Research Institute, San Francisco, CA, USA*
[b]*Department of Computer Engineering, University of California, Santa Cruz, CA, USA*

Abstract. Urban intersections are the most dangerous parts of a blind or visually impaired pedestrian's travel. A prerequisite for safely crossing an intersection is entering the crosswalk in the right direction and avoiding the danger of straying outside the crosswalk. This paper presents a proof of concept system that seeks to provide such alignment information. The system consists of a standard mobile phone with built-in camera that uses computer vision algorithms to detect any crosswalk visible in the camera's field of view; audio feedback from the phone then helps the user align him/herself to it. Our prototype implementation on a Nokia mobile phone runs in about one second per image, and is intended for eventual use in a mobile phone system that will aid blind and visually impaired pedestrians in navigating traffic intersections.

Keywords: Blindness, visual impairments, navigation, computer vision, cell phone, mobile phone, camera phone

1. Introduction

Many urban traffic and pedestrian accidents occur at intersections (with about 22% of all fatal pedestrian accidents happening at these locations in the US [19]), which are especially dangerous for blind or visually impaired pedestrians, for whom intersections are generally agreed to be the most difficult and risky aspects of independent travel. Currently, blind travelers must listen to the traffic and attempt alignment with the sounds of vehicles receding into the distance. The sound of traffic starting to move parallel to one's direction of travel is used to deduce that the traffic light has turned green (and to assume that the "Walk" signal has appeared). These techniques often require waiting through one or more traffic light cycles to reduce the chances of choosing the incorrect time to cross.

Several types of Audible Pedestrian Signals (APS) have been developed to assist blind and visually impaired individuals in knowing when to cross intersections [2–4]. However, while widespread in some countries, their adoption is very sparse in others (such as the US, where they are completely absent in most cities).

Some technology addresses the broader navigation aspect of the travel task [1]. For example, Talking Signs® [8,18] allows blind travelers to locate and identify landmarks, signs, and facilities of interest, at intersections and other locations. Signals from installed infrared transmitters are converted to speech by a receiver carried by the traveler. This has been found to enhance safety, efficiency and knowledge about the intersection [9]. A number of related technologies have been proposed, and such systems are spreading, but are still only available in very few places. Finally, GPS (Global Positioning System) technology has the advantage of being widely available [11,16,17], but unfortunately on urban sidewalks near buildings the localization accuracy is only sufficient to resolve within a few street addresses – not nearly enough to provide useful guidance within a traffic intersection.

The alternative approach that we propose in this paper is to use a portable computer vision-based system to identify important features in an intersection. With this system, the user takes an image of the intersection with a digital camera, which is analyzed by software

*Address for correspondence: J. Coughlan, The Smith-Kettlewell Eye Research Institute, 2318 Fillmore St., San Francisco, CA 94115, USA. E-mail: Coughlan@ski.org.

ISSN 1055-4181/08/$17.00

run on a computer connected to the camera, and the output of the software is communicated to the user with synthesized speech or acoustic cues. The great strength of the computer vision approach is that it offers virtually unlimited potential for acquiring information about one's surroundings, but does not require any infrastructure beyond what is already provided for sighted people – namely, street signs, traffic lights and painted street markings.

This paper presents a prototype computer vision system for addressing one specific aspect of the intersection problem: the challenge of aligning oneself correctly with the crosswalk, which is crucial for entering the crosswalk in the right direction and avoiding the danger of straying outside the crosswalk. While a comprehensive assistive technology aid for dealing with intersections will need to incorporate additional functionality (such as determining the layout of the intersection, the location and timing of the pedestrian signals, etc.), a solution to the alignment problem will help significantly with the problem of *how to cross the intersection once the traveler has arrived there.* Indeed, this problem has been identified in our consultations with blind individuals as more important than the *detection* of traffic intersections, which is typically achieved by standard orientation and mobility techniques.

In previous work [5] we developed a novel computer vision algorithm (implemented on a desktop computer) to detect and locate crosswalks in an image, which is robust to real-world conditions such as cluttered scenes with shadows, saturation effects, slightly curved stripes and occlusions. We have simplified our previous algorithm so that it requires significantly less computational power. These improvements have allowed us to implement a crosswalk detection algorithm on an off-the-shelf Nokia camera (mobile/cell) phone that runs in about one second per image (see Fig. 9). The functionality we have implemented is intended for eventual integration in a mobile phone system that will aid blind and visually impaired pedestrians in navigating traffic intersections.

2. Computer vision approach

This section first explains why we have chosen the mobile phone hardware platform for our system, and then describes our computer vision algorithms.

2.1. Suitability of the hardware platform

We have chosen the mobile phone as the platform for our assistive technology device because it combines a computer and digital camera in a small, low-cost, multi-function package. Computer vision software can be installed and run on an off-the-shelf mobile camera phone without any hardware modification of any kind (and without interfering with the usual cell phone functionality). Compared with nearly all other past, present or proposed assistive technology aids for visual impairments, the mobile phone framework also has the advantage of not requiring the user to carry any extra device or hardware. We also note that the mobile phone is a mainstream consumer product which raises none of the cosmetic concerns that might arise with other assistive technology requiring custom hardware. Finally, the mobile phone contains a loudspeaker and vibrator, which can provide audio and tactile information to the user, as well as text-to-speech and limited voice recognition capability.

However, compared to a standard desktop PC typically used to implement computer vision algorithms, the computer in a mobile phone has much less computational power. Indeed, our preliminary experiments show the cell phone used in our prototype system (the Nokia 6681, shown in Fig. 9) to be an order of magnitude slower than a standard PC for integer arithmetic calculations (implemented in Symbian C++, which is the most efficient language available for this platform). Because mobile phones lack a floating point processing unit (FPU), they are even slower for floating-point calculations (e.g. non-integer multiplication and division), which are used extensively in computer vision.

The limited power of the mobile phone CPU imposes a strong constraint on the computational demands of our computer vision algorithms – forcing us to choose a problem that is small enough to be tractable but still worth solving, and to devise extremely efficient algorithms. We believe that the prototype we describe in this paper, which takes only a second to analyze each image, demonstrates that this constraint has been satisfied.

Even if the mobile phone CPU is powerful enough to implement computer vision algorithms, however, one may still question the practicality of an assistive technology aid that requires blind and visually impaired users to take photographs. Fortunately, our experience with blind subjects demonstrates their ability to take images of sufficient quality.

Fig. 1. Typical intersection images photographed by completely blind subject (zebra crosswalks in left and center images, two-stripe crosswalk in right image).

Figure 1 shows typical images of traffic intersections photographed by a completely blind subject using a digital camera. The subject was first given a brief training session advising him to hold the camera as steadily, and as closely to the horizontal, as possible. He was instructed to walk to the corner of several intersections and to take digital photographs he judged to be aligned with the crosswalks. The only help given to the subject was the presence of sighted experimenters at close range to intervene if necessary to prevent accidents. Clearly, the subject was able to find the crosswalks and take clear photographs of them. The main defect of the photographs is that the view of the crosswalk was often misaligned – the very problem that our system is designed to address.

Finally, while our mobile phone prototype is not yet ready for testing by blind subjects, we have performed experiments demonstrating their ability to use a mobile camera phone. Such experiments are important because the cell phone camera is more susceptible to motion blur and harder to aim than the higher-quality, dedicated digital camera used in Fig. 1. The experiments tested the ability of blind subjects to use a mobile phone-based wayfinding system to find landmarks of interest that were labeled by signs bearing special printed patterns, designed to be located and read by the mobile phone from a distance [6,7]. The results showed that the subjects were able to use the wayfinding system effectively, which demonstrates that they were able to take images of adequate quality with the mobile phone.

2.2. Related work in computer vision

Little work has been done on computer vision algorithms for detecting crosswalks for the blind and visually impaired [20,21,23,24]. This body of work has focused on the zebra (striped) crosswalk (see Fig. 1a, b), which is a common type of crosswalk. In this paper we will also specialize to the zebra crosswalk, and will save the more difficult problem of finding the two-stripe crosswalk (Fig. 1c), another common type of crosswalk (that is much less visible than the zebra crosswalk), for later research. We note that the yellow color of the zebra crosswalks we have encountered in our neighborhood of San Francisco is a distinctive color that could be used to guide the detection process. However, we have avoided using color as a detection cue because many zebra crosswalks elsewhere are white, and we would like our algorithm to detect zebra crosswalks of any color.

Existing algorithms for detecting zebra crosswalks have been demonstrated on fairly simple images in which the Hough transform, which is a standard method for finding straight lines that extend across a large portion of an image [10], is typically sufficient for extracting the borders of the crosswalk stripes. However, in real-world conditions, such as cluttered scenes with shadows, saturation effects, slightly curved stripes, crosswalk paint irregularities and occlusions, the Hough transform is often inadequate as a preprocessing step.

2.3. Figure-ground segmentation

Instead of relying on a tool for grouping structures globally such as the Hough transform, we use a more flexible, *local* grouping process based on figure-ground segmentation. Unlike the Hough transform, this process is robust to local deformations of straight lines, such as those due to curvature of the road, cracks in the pavement and paint imperfections, that can break the long edge of a crosswalk stripe into disjoint pieces.

Figure-ground segmentation is a process of grouping a collection of visual elements into two sets, the “figure” (foreground) and the “ground” (background), according to compatibility relationships among elements. Elements that satisfy these compatibility relationships tend to be grouped into the figure, while the elements that violate the compatibility relationships tend to be grouped into the ground. Historically, figure-ground segmentation was originally used to describe the grouping of image features by the human visual system using generic, Gestalt-type, compatibility relationships such as collinearity, parallelism and common color.

To solve specific problems such as detecting straight lines or crosswalks, we build on work in computer vision on *object-specific* figure-ground segmentation [15, 25], which tailors the figure-ground process to a particular domain of interest. For our application, the elements to be segmented into figure or ground are straight-line edge segments (see Fig. 2), which are extracted from a simple edge map of an image by finding connected groups of edge pixels lying along a straight line. (An edge map is a basic tool in computer vi-

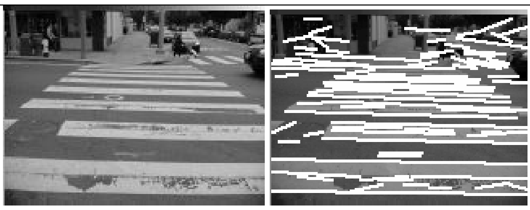

Fig. 2. Left, typical intersection image photographed by camera phone. Right, straight-line segments extracted from image shown in white.

sion that provides an estimate of the likely locations of boundaries in the image, which are signaled by discontinuities in image intensity that typically occur at transitions from one region to another [10]). The edge map is calculated on the image taken by the Nokia 6681 at coarse resolution (160 $\times$ 120 pixels), after converting it to grayscale and performing some simple image blurring (to reduce the number of spurious edges).

Straight-line edge segments belonging to a zebra crosswalk satisfy several compatibility relationships that may be used to segment them into the figure (and to distinguish them from the background). The segments are relatively parallel to one another, and segments belonging to consecutive stripes are close together, compared to the size of the image (see Fig. 2b). Consecutive segments also have opposite edge polarities. (The polarity of an edge indicates which side of the edge has the brighter intensity – for instance, a yellow stripe is brighter than the pavement adjacent to it.) Finally, the higher up in the image a stripe appears, the narrower its apparent width (since the stripes, and gaps between them, have uniform width in 3-D, and stripes that appear higher in the image are farther from the camera).

In order to implement a figure-ground segmentation process as an algorithm that enforces these compatibility relationships, we use a tool from machine learning, the graphical model [14]. The strength of the graphical model is that it models compatibility relationships statistically, in a way that incorporates contextual information by accumulating evidence and postponing making decisions (e.g. grouping into figure or ground) until enough evidence has been considered. This statistical property of the graphical model is important because it avoids the problem of making premature decisions based on hard and fast rules. For example, among a group of segments that belong to the crosswalk, a few segments may be less parallel to each other than usual (e.g. because of defects in the crosswalk); the graphical model figure-ground algorithm may nevertheless successfully group them as figure because of overwhelming support from enough nearby crosswalk segments that are more compatible with each other.

For details on the graphical model approach that we employ, see [22]; the approach is a simplification of our earlier graphical model algorithm for finding crosswalks [5], which was more computationally intensive. The main idea of the approach is to generate hypothetical groupings of small numbers of elements, which we call "factors," and to calculate the likelihood that any element belongs to figure by integrating the evidence from all the factors that it belongs to. Each factor is evaluated in terms of the compatibility relationships of its members, resulting in an overall compatibility score; the more factors an element belongs to with high compatibility scores, the higher the likelihood that the element belongs to figure.

Specifically, in our application a factor is a group of three consecutive segments that satisfy the compatibility relationships described earlier: parallelism, proximity, alternating edge polarity, and the constraint that stripes higher in the image appear narrower (the "monotonicity" constraint) because they are farther from the camera. Here proximity means that (a) the gap between consecutive segments should be sufficiently small, and that (b) the segments should have significant horizontal overlap. Factors are obtained by searching the segments in the image for suitable groupings of three segments; only a group of three segments that satisfies the proximity, polarity and monotonicity constraints are chosen to be a factor. For each factor thus obtained, its compatibility score is defined by summing two individual compatibility scores, one for proximity (as defined by overlap) and the other for parallelism. Finally, for each element, the compatibility scores of all the factors it belongs to are summed; if the total is higher than a certain threshold then the element is assigned to the figure.

We now define the factors in equations. Given three segments i_1, i_2, i_3, there are two individual compatibility scores: $f_1(i_1, i_2, i_3)$, which rewards horizontal overlap among the segments, and $f_2(i_1, i_2, i_3)$, which rewards parallelism. We define $f_1(i_1, i_2, i_3) = \min\{50, \min[O(i_1, i_2), O(i_2, i_3)]\}$ where $O(i_1, i_2)$ is the amount of overlap between segments i_1 and i_2, namely, the number of pixel columns (i.e. vertical slices) that intersect both segments i_1 and i_2. Note that $f_1(i_1, i_2, i_3)$ is guaranteed to be within the range 0 to 50. Next, $f_2(i_1, i_2, i_3) = 50 - \min\{50, 1000 \max[|m(i_1) - m(i_2)|, |m(i_2) - m(i_3)|]\}$, where $m(i_1)$ is the slope of segment i_1 and $|m(i_1) - m(i_2)|$ is the absolute value of the difference of the slopes of segments i_1

Fig. 3. Output of detection algorithm applied to image in previous figure.

andi_2 (which equals 0 if the segments are perfectly parallel). The multiplier of 1000 in front of the *max* term is a heuristic that makes the strength of $f_2(i_1, i_2, i_3)$comparable to that of $f_1(i_1, i_2, i_3)$. We see that $f_2(i_1, i_2, i_3)$ is also guaranteed to be within the range 0 to 50. The total compatibility score of the factor containing segments i_1, i_2, i_3 is $F(i_1, i_2, i_3) = f_1(i_1, i_2, i_3) + f_2(i_1, i_2, i_3)$. For each segment i, the total evidence that i belongs to figure is then the sum of all factors F that contain segment i.

The result of the algorithm on the image in Fig. 2a is given in Fig. 3, which shows the segments from Fig. 2b that are grouped into the figure – i.e. all segments that have total evidence above a particular threshold. Note that most of the segments that belong to the crosswalk are grouped into the figure, and almost all the other segments in the image are discarded. (More results are shown in Sec. 3.)

2.4. Extracting geometric information

Thus far we have focused on developing an algorithm to automatically detect and locate crosswalks. We now address what this algorithm is used for, which is to measure the user's orientation to the crosswalk, and to determine what direction the user needs to turn to align him/herself properly. Techniques from 3-D perspective geometry [10] are used to perform these calculations, which we outline briefly in this subsection.

The basic orientation measure that needs to be determined is the angle between the camera's line of sight and the direction indicated by the crosswalk stripes. The camera's line of sight direction corresponds to the center of the image, and the direction of the crosswalk is identified by the *vanishing point* (see Fig. 4) of the path defined by the two borders of the crosswalk (which may, in general, lie inside or outside the image frame). This vanishing point is the direction that the user wants to walk to traverse the crosswalk, and we denote it by D. If the camera is held horizontally, a simple formula expresses the orientation of the user relative to the crosswalk in terms of the focal length of the camera (which is a fixed quantity that is a property of the camera lens) and the location of D in the image.

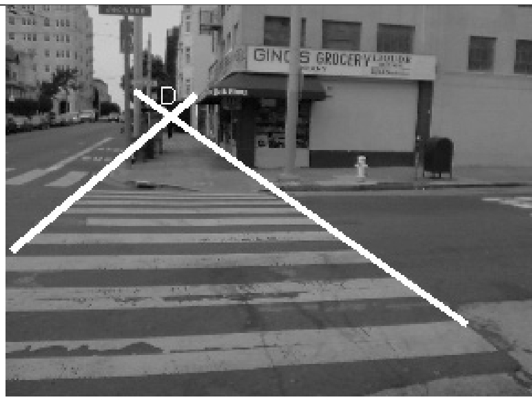

Fig. 4. Vanishing point, labeled D, is the intersection of the white lines defined by the two borders of the crosswalk. Since D is slightly left of the center of the image (by an angle that we can compute), the user needs to turn slightly left to align him/herself properly to the crosswalk.

The vanishing point D can be estimated even if one or both of the two crosswalk borders are not clearly visible (see Fig. 5). First, the horizon line is determined from any three consecutive zebra stripe edges (using knowledge of the fact that the stripes, and gaps between them, have uniform width in 3-D) using the *cross ratio*, which is a property of any four collinear points that is invariant to the perspective from which the points are viewed. For any vertical slice through the three consecutive zebra stripes (which intersect the slice in points A, B and C), the cross ratio can be used to determine the fourth point, Z, which is the intersection of the horizon with this slice given points A, B and C. A second vertical slice then gives an additional point on the horizon, thus determining the horizon line.

Next, the vanishing point D' of the long edges of the zebra stripes is determined; these stripe edges point perpendicularly to the desired vanishing point direction D, and their vanishing point D' also lies on the horizon. D can then be calculated from $D'$and the horizon.

These calculations can be performed even if the camera is tilted from the horizontal (which is often the case when a blind subject takes a photograph): the orientation of the horizon determines the amount of tilt, and

Fig. 5. Determining the horizon and vanishing point when only one of the two borders of the crosswalk is visible. The three solid white lines are the crosswalk edges used to estimate the horizon (dashed white line) and the vanishing point (white dot on horizon line, labeled *D*.

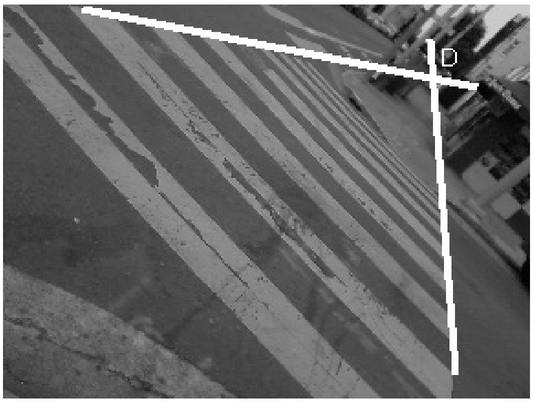

Fig. 6. Severely tilted crosswalk image could be misleading. Vanishing point, labeled *D*, is on the right side of the image even though the camera is aligned to the crosswalk.

this can be compensated for by (virtually) rotating the image so that the horizon is horizontal. The ability to compensate for the tilt is important, since otherwise the location of the vanishing point D might be misinterpreted, as shown in Fig. 6. In this image, the camera is severely tilted from the horizontal. However, if the tilt is compensated for by rotating the image about its center so that the horizon (and thus the stripes) is horizontal, D will lie directly above the center. This shows that the camera is aligned to the crosswalk, even though it is tilted.

So far this functionality has only been tested on a desktop computer, but it will be straightforward to implement on the camera phone when we are ready to incorporate it into the system's user interface. A small number of calculations are required for this functionality – only on the order of tens of floating point calculations, which is negligible compared to what is demanded by the crosswalk detection algorithm that precedes it (e.g. creating the edge map alone requires performing calculations on each of the tens of thousands of pixels in the image).

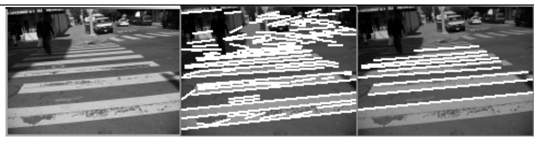

Fig. 7. Sample result of mobile phone algorithm. Left to right: original image taken by mobile phone; segments extracted from image, shown in white; and segments selected by algorithm as belonging to the crosswalk, also in white.

3. Experimental results

Figures 7 and 8 show experimental results of our mobile phone system, demonstrating the ability of our algorithm to detect and locate zebra crosswalks in a variety of images. These images show crosswalks taken at different angles, with varying amounts of defects in the paint, and with occlusions partly obscuring the crosswalk due to objects such as vehicles. The same exact algorithm was used for each image. Figure 9 shows the mobile phone system displaying a crosswalk (the display is for debugging purposes and is obviously of no use to blind users).

The results are of slightly lesser quality than those obtained by our earlier crosswalk algorithm [5] that was implemented on a desktop. In particular, some of the crosswalk stripe edges are missed by the algorithm, especially the one that is on the leading edge of the stripe nearest the camera. Also, some of the edges output by the algorithm erroneously extend beyond the borders of the crosswalk. However, we stress that our new algorithm is preliminary, and note that the results are of high enough quality to permit the estimation of the user's orientation relative to the crosswalk.

Moreover, we are planning to remedy these defects in future research. The reason the leading edge is missed by the algorithm is that it has no evidence to support it from segments below, only from the segments above; this could be corrected by boosting the evidence given to the segment that is lowest in the image. Also, we can run a post-processing procedure to search for edges perpendicular to the long edges (which define the vanishing point D as in Fig. 4) so that we can define the extent of the crosswalk area, and crop out any edges that extend beyond this area.

Fig. 8. More results of camera phone algorithm.

Fig. 9. A crosswalk shown as it appears on the display of the Nokia 6681 mobile phone.

4. Conclusion

We have demonstrated a novel mobile phone-based system for finding zebra crosswalks. The purpose of our system is to help blind and visually impaired pedestrians align themselves properly to a crosswalk before crossing. A prototype system has been implemented on the Nokia 6681 mobile phone, which detects the crosswalk stripes in an image in about one second. The success of this prototype demonstrates the feasibility of implementing real-time computer vision algorithms on the mobile phone despite its limited computational power (relative to the desktop computer typically used for such algorithms).

This paper describes our work on crosswalk detection and alignment issues up to the time of submission. Subsequent work to ready the prototype for testing by blind and visually impaired subjects is underway [12,13] and will be published in a future journal article. This work will include several components: (a) testing the algorithm on a dataset of images photographed by subjects using a camera phone, and refining it to improve its accuracy and robustness; (b) deciding which information is most important to present to the blind user; and (c) developing a suitable user interface that allows the user to easily acquire images, and which communicates information to the user efficiently, through some combination of audio tones and synthesized speech.

After these steps have been taken, subject testing will be conducted to determine whether the system is usable, measure its performance quantitatively and elicit feedback to see how the system can be improved. At a later stage, the system will also be augmented by additional functionality, including detecting the two-stripe as well as the zebra crosswalk; detecting "Walk" lights, pedestrian signals and/or traffic lights; and monitoring the pedestrian's path to prevent him/her from straying out of the crosswalk while traversing it.

Acknowledgments

The authors were supported by the National Institute on Disability and Rehabilitation Research (grant number H133G030080) and the National Eye Institute (grant number EY015187-01A2). We would like to thank Dr. Josh Miele, Tom Fowle and Bill Gerrey for many useful discussions.

References

[1] A. Arditi and J.A. Brabyn, Signage and wayfinding, in: *The Lighthouse Handbook on Vision Impairment and Vision Rehabilitation*, (2 Vols.), Silverstone, Lang, Rosenthal & Faye, eds., Oxford University Press, 2000.

[2] J.M. Barlow, B.L. Bentzen and L. Tabor, Accessible pedestrian signals: Synthesis and guide to best practice, *National Cooperative Highway Research Program*, 2003.

[3] J.M. Barlow, B.L. Bentzen and T. Bond, Blind pedestrians and the changing technology and geometry of signalized intersections: Safety, Orientation, and Independence, *Journal of Visual Impairment and Blindness* **99** (2005), 587–598.

[4] B.L. Bentzen, J.M. Barlow and D. Gubbé, Locator tones for pedestrian signals, *Transportation Research Record* **1705** (2000), 40–42.

[5] J. Coughlan and H. Shen. A Fast Algorithm for Finding Crosswalks using Figure-Ground Segmentation, in: *Proc 2nd Workshop on Applications of Computer Vision*, in conjunction with ECCV 2006, Graz, Austria, May 2006.

[6] J. Coughlan and R. Manduchi, Functional Assessment of a Camera Phone-Based Wayfinding System Operated by Blind Users, *IEEE-BAIS* (*IEEE Computer Society and the Biological and Artificial Intelligence Society*) *RAT-07* (*Research on Assistive Technologies*) *Symposium*, Dayton, Ohio, April 2007.

[7] J. Coughlan, R. Manduchi and H. Shen, Cell Phone-based Wayfinding for the Visually Impaired, In Proc, *1st International Workshop on Mobile Vision*, in conjunction with European Conference on Computer Vision (ECCV) 2006, Graz, Austria, May 2006.

[8] W. Crandall, B.L. Bentzen, L. Myers and J. Brabyn, New orientation and accessibility option for persons with visual impairment: transportation applications for remote infrared audible signage, *Clinical and Experimental Optometry* **84**(3) (May 2001), 120–131.

[9] W. Crandall, J. Brabyn and B.L. Bentzen, Remote Infrared Signage Evaluation for Transit Stations and Intersections, *Journal of Rehabilitation Research and Development* **36**(4) (1999), 341–355.

[10] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, 2002.

[11] J. Fruchterman, Talking Maps and GPS systems, *Paper presented at the Rank Prize Funds Symposium on Technology to Assist the Blind and Visually Impaired*, Grasmere, Cumbria, England, March 25–28, 1996.

[12] V. Ivanchenko, J. Coughlan and H. Shen, *Crosswatch: a Camera Phone System for Orienting Visually Impaired Pedestrians at Traffic Intersections*, To appear in 11th International Conference on Computers Helping People with Special Needs (ICCHP '08), Linz, Austria, July, 2008.

[13] V. Ivanchenko, J. Coughlan and H. Shen, *Detecting and Locating Crosswalks using a Camera Phone*, To appear in the Fourth IEEE Workshop on Embedded Computer Vision, in conjunction with Computer Vision and Pattern Recognition (CVPR '08), Anchorage, Alaska. June, 2008.

[14] M. Jordan, *Learning in Graphical Models*, Cambridge MA: MIT Press, 1999.

[15] S. Kumar and M. Hebert, *Man-Made Structure Detection in Natural Images using a Causal Multiscale Random Field*, CVPR 2003.

[16] J. Loomis, R. Klatzky and R. Golledge, Navigating without vision: basic and applied research, *Optom Vis Sci* **78**(5) (May 2001), 282–289.

[17] J.M. Loomis, R.L. Golledge, R.L. Klatzky, J. Speigle and J. Tietz, *Proceedings of the First Annual International ACM/SIGCAPH Conference on Assistive Technologies*, Marina Del Rey, California, Oct 31-Nov 1, 1994, New York: Association for Computer Machinery, 85–90.

[18] W. Loughborough, Talking Lights, *Journal of Visual Impairment and Blindness*, 1979, 243.

[19] Pedestrians: Types of Problems Being Addressed, Document available from the National Cooperative Highway Research Program at http://safety.transportation.org/htmlguides/peds/.

[20] S. Se and M. Brady, Road Feature Detection and Estimation, *Machine Vision and Applications Journal* **14**(3) (July 2003), 157–165.

[21] S. Se, *Zebra-crossing Detection for the Partially Sighted*, In CVPR 2000, South Carolina, June 2000.

[22] H. Shen and J. Coughlan, Grouping Using Factor Graphs: an Approach for Finding Text with a Camera Phone, To appear in *Workshop on Graph-based Representations in Pattern Recognition* (GbR '07, associated with The International Association for Pattern Recognition), June 2007, Alicante, Spain.

[23] M.S. Uddin and T. Shioyama, *Bipolarity- and Projective Invariant-Based Zebra-Crossing Detection for the Visually Impaired*, 1st IEEE Workshop on Computer Vision Applications for the Visually Impaired, CVPR 2005.

[24] S. Utcke, Grouping based on Projective Geometry Constraints and Uncertainty, in: *Int'l Conference on Computer Vision '98*, Bombay, India Jan, 1998.

[25] S.X. Yu and J. Shi, Object-Specific Figure-Ground Segregation, *Computer Vision and Pattern Recognition 2003*, Madison, WI, June 2003.