SMARTPHONE-BASED CROSSWALK DETECTION AND LOCALIZATION FOR VISUALLY IMPAIRED PEDESTRIANS

Vidya N. Murali and James M. Coughlan

The Smith-Kettlewell Eye Research Institute
San Francisco, CA
{vidya, coughlan}@ski.org

ABSTRACT

This paper describes recent work on the "Crosswatch" project [7], which is a computer vision-based smartphone system developed for providing guidance to blind and visually impaired travelers at traffic intersections. A key function of Crosswatch is self-localization – the estimation of the user's location relative to the crosswalks in the current traffic intersection. Such information may be vital to users with low or no vision to ensure that they know which crosswalk they are about to enter, and are properly aligned and positioned relative to the crosswalk. However, while computer vision-based methods have been used [1,9,14] for finding crosswalks and helping blind travelers align themselves to them, these methods assume that the entire crosswalk pattern can be imaged in a single frame of video, which poses a significant challenge for a user who lacks enough vision to know where to point the camera so as to properly frame the crosswalk.

In this paper we describe work in progress that tackles the problem of crosswalk detection and self-localization, building on recent work [8] describing techniques enabling blind and visually impaired users to acquire 360° image panoramas while turning in place on a sidewalk. The image panorama is converted to an aerial (overhead) view of the nearby intersection, centered on the location that the user is standing at, so as to facilitate matching with a template of the intersection obtained from Google Maps satellite imagery. The matching process allows crosswalk features to be detected and permits the estimation of the user's precise location relative to the crosswalk of interest. We demonstrate our approach on intersection imagery acquired by blind users, thereby establishing the feasibility of the approach.

***Index Terms*—** Computer vision, blindness, visual impairment, traffic intersection, self-localization

1. INTRODUCTION AND RELATED WORK

Traffic intersections are among the most dangerous places that blind and visually impaired travelers encounter. Standard orientation and mobility techniques (such as using the white cane to determine the location of the curb cut adjoining a crosswalk, or listening to traffic sounds to infer the timing of the traffic lights) provide valuable information [15], and at some intersections accessible pedestrian signals (APS) [4] provide walk light timing information in an audible form. Unfortunately, much important information about the layout of the intersection, the timing of the traffic lights, and the traveler's alignment and orientation relative to a desired crosswalk may be difficult or impossible for him/her to determine without vision.

One technology that is very useful for self-localization at traffic intersections and other locations is GPS. While GPS localization accuracy is sufficient to determine which traffic intersection the traveler is standing at (and in some cases even which corner of the intersection), it is limited in urban settings, near buildings, trees and bus stops, where errors due to reflections and signal dropouts prevent accuracy being better than within a few street addresses [6]; thus GPS cannot offer direct assistance in knowing where it is safe to cross.

Computer vision is a natural choice of technology to infer the presence and location of crosswalks in the traveler's environment, since this function nicely complements the information provided by GPS. There is a variety of work (e.g., [16]) on computer vision-based self-localization, including a system [11] that infers location based on the detailed appearance of the skyline and a real-time system [3] that infers location from panoramic images on mobile devices. However, such systems require the use of detailed 3D models of urban environments, which are complex, memory-intensive and may be difficult to acquire.

Instead of relying on a complex 3D model we chose to harness a simple and readily available 2D model of the urban environment based on satellite imagery such as can be obtained freely from Google Maps. In addition to its simplicity, the 2D image model of the intersection has the added advantage of providing metric information relative to the *specific features that matter to a blind or visually impaired traveler* – namely, crosswalk marking patterns. By contrast, a 3D model consisting of features extracted from buildings, vegetation and other environmental structures is likely to contain few crosswalk features, which necessitates additional information to locate the crosswalk features relative to the model.

Past work on the Crosswatch project [7,8,9,10] and the work of Ahmetovic et al. [1] also analyze 2D images to detect and locate crosswalks, but these projects have the limitation that they analyze images one at a time. This limitation forces the user to capture the entire crosswalk of interest in a single video frame, which can be challenging for users who don't have enough vision to know where to point the camera.

To overcome this limitation, we analyze an entire 360° image panorama, acquired on the smartphone, which does not require the user to point the camera accurately. The acquisition of the panorama is facilitated by real-time tactile feedback to help the user hold the camera horizontally even without being able to see the image in the camera viewfinder. The image panorama is then warped into an aerial (overhead) view of the intersection near the user, which can be matched to a template of the intersection obtained from satellite imagery. The match can then be used to verify the presence of the crosswalk(s) in the panorama and to determine the location of the user relative to the crosswalk.

Finally, we mention that related work (e.g., [2, 10]) has also been done on using computer vision to detect and report the status of traffic signals such as walk lights and traffic lights, which has great potential for use in the great majority of intersections that lack accessible pedestrian signals.

2. OVERALL APPROACH

Self-localization requires imagery taken by the user to be matched to a template of the traffic intersection. (We assume that GPS has sufficient resolution to determine the intersection that the user is standing at, but not necessarily which corner, and certainly not the user's precise location relative to the crosswalks.) The template for this specific intersection is then used for self-localization. In this section we provide an overview of how this is done.

First, a full 360° image panorama is acquired by the smartphone user, using a simple app that was developed previously [8] to acquire a sequence of VGA images, such that one image is taken roughly every 20° of bearing as the user stands in place and rotates clockwise. Tactile feedback is issued to warn the user any time the camera is held sufficiently far from horizontal. The accelerometer and magnetometer measurements are saved for each image in the sequence, and these measurements are used to estimate bearing (i.e., azimuth relative to magnetic north) and camera pose.

The image sequence is then stitched into a 360° cylindrical panorama [13] offline using commercially available stitching software. Next we model the traffic intersection near where the user is standing as a ground plane that is horizontal (relative to gravity); this model is approximately accurate even when the streets adjoining the intersection are on a hill, since the intersection area itself is constructed to be as horizontal as possible. The approximate height of the camera from the ground is measured for each user (and is fairly constant for each person). Knowledge of camera height, in conjunction with the accelerometer and magnetometer readings for images in the panorama sequence, allows us to unwarp the panorama into an aerial (overhead) view image, which is centered at the estimated location of the user's feet. The magnetometer readings also determine the bearing of the camera for each image; the bearing is measured relative to magnetic north but can easily be translated relative to geographic north using knowledge of the local magnetic declination. The aerial image is orientation normalized by rotating it such the "up" direction in the image is aligned to geographic north. Finally, this procedure yields an aerial image whose scale is determined (e.g., in meters per pixel).

The aerial image must be matched to a template of the traffic intersection to determine the user's current location in the intersection. For each intersection of interest, this template is obtained from Google Maps satellite images by manually segmenting out the crosswalk stripes; in the future manual segmentation can be replaced by an automated procedure and/or the use of crowd-sourcing, which will make it practical to obtain templates for many intersections. (Crowd-sourcing has already been successfully applied to the related problem of providing information about a visually impaired person's surroundings, as is performed with the "VizWiz" iPhone app [5].) Like the final aerial image, the template image is oriented such that the "up" direction in the image points toward geographic north, and the scale of the image is specified (in meters per pixels), which facilitates direct comparison between the aerial image and the template.

Our approach should apply to any standard crosswalk marking pattern, and should work even in intersections without crosswalks in which a single limit line is painted for each road entering the intersection (indicating where cars should stop). However, as this is work in progress we decided to apply it first to the "two-stripe" (two transverse lines) crosswalk marking, shown in Fig. 1, which is a type of crosswalk that is both more common and less visible than the "zebra" (continental) crosswalk that has received more emphasis in past work on computer vision-based crosswalk detection (e.g., [14]).

Fig. 1. The "two-stripe" (two transverse lines) crosswalk marking pattern.

Finally, we note that our current system prototype performs all computer vision processing offline, but in the future we will develop a system that runs entirely on the smartphone. This system will provide real-time feedback where necessary (e.g., in alerting the user when the camera is no longer held horizontal) and will respond with acceptable delays for other types of information; for instance, after the panorama is acquired, the computer vision processing could be done on the smartphone CPU or uploaded to the cloud for self-localization results in several seconds or less.

3. ALGORITHM DETAILS AND EXPERIMENTAL RESULTS

3.1. Image panorama acquisition

Following the procedure reported in [8], a blind or visually impaired user acquires an image panorama on a smartphone by standing in one place and slowly rotating clockwise (as viewed from above) as the system acquires a new VGA image approximately every 20° of bearing, saving a total of 18 images for one panorama. We created an Android app to perform this function [8], with audio feedback to signal the acquisition of each image and to signal the successful completion of an entire 360° panorama. In the app, the current accelerometer and magnetometer readings are sampled several times per second.

A simple accelerometer calibration procedure was performed before conducting the user experiments to increase the accuracy of the accelerometer readings by compensating for constant gain and offset parameters that slightly distort the raw x,y,z accelerometer components. (Currently we are not calibrating the magnetometer as well but may use such a procedure in the future.) Given the accelerometer and magnetometer readings, it is straightforward to estimate the three Euler angles (roll, pitch and yaw) for the smartphone, where the yaw (i.e., azimuth or bearing) is measured relative to magnetic north, and roll and pitch are measured relative to the horizontal plane (as defined by gravity).

To help the user hold the camera horizontal during the image acquisition process, we alert the user with tactile feedback (using the smartphone vibrator) any time the roll or pitch of the camera deviates sufficiently far from zero (i.e., horizontal); this user interface feature allows the user to vary only the camera yaw (i.e., bearing) without deviating significantly from horizontal. The feature is especially useful for totally blind users who may not otherwise know how horizontally they are holding the camera.

We recruited two completely blind volunteers to participate in our study, which was begun previously and reported in [8]. After following the IRB consent process, each volunteer was given a simple training procedure to acquaint him/her with the purpose of the study, the proper way of holding the smartphone and how to use the smartphone panorama app. Each volunteer practiced using the app indoors, where the experimenters could intervene in any way necessary to help him/her use the app properly. Then each volunteer was guided to two nearby (four-way) traffic intersections, where he/she was instructed to acquire a single panorama sequence for each corner of the intersection. Thus a total of eight panorama sequences were sampled by each volunteer; once the volunteer was handed the smartphone to acquire a panorama, the experimenters provided no assistance (but were ready to intervene if necessary to ensure the volunteer's safety).

3.2. Creation of image panorama

We used Microsoft ICE (Image Composite Editor, http://research.microsoft.com/en-us/um/redmond/groups/ivm/ice/), a desktop application for performing automatic panoramic stitching (i.e., image mosaicking). Given the input images, the software automatically constructed a 360° cylindrical panorama for each image sequence (see Fig. 2 for an example). In addition to outputting the panorama as a single image, the software also provided relative estimates of the camera bearing (i.e., azimuth or yaw) for each image in the sequence, relative to the -180° to +180° range encompassed in the width of the panorama. Given the accelerometer and magnetometer readings for the first image in a panorama sequence, and the magnetic declination, the resulting absolute bearing estimate (relative to magnetic north) of the first image was used to estimate the bearing (relative to geographic north) of every pixel column in the panorama.

Fig. 2. An image panorama constructed using imagery acquired by a blind user. Notice the presence of two crosswalks (the "two-stripe" type), which can be seen in the aerial view (Fig. 3) to meet at roughly right angles.

Despite the use of tactile feedback to help the user hold the camera horizontal during the image acquisition, the horizon line (defined by gravity) was not precisely aligned to the center row of the panorama. Thus we used the accelerometer and magnetometer readings from all the images in the sequence to estimate the horizon row for each pixel column in the panorama. As a result, we were able to determine the pitch (relative to the horizon defined by gravity) and yaw (relative to geographic north) for every pixel in the panorama image.

In addition to capturing crosswalk patterns that may not easily fit in a single camera image, we note that an additional benefit of the image panorama is that the image mosaicking process removes a variety of moving occluders in the image [13], such as moving vehicles and pedestrians, thereby improving the likelihood that the crosswalk features of interest are visible.

3.3. Reconstruction of aerial view

We modeled the ground near the user's feet as a horizontal plane parameterized by an *(a,b)* coordinate system, where the origin *(0,0)* is at the user's feet. (This model is reasonably accurate, but of course any objects lying on or protruding from the ground, such as vehicles, pedestrians or fire hydrants, violate the assumption that the entire neighborhood lies on the ground plane.) Given the height of the camera, for any point *(a,b)* on the plane it is straightforward to determine the pitch and yaw relative to the camera center. (We represent the coordinates *(a,b)* in units of meters.) The pitch and yaw in turn determine the corresponding pixel location in the panorama, which means that we can map the intensity of the scene panorama in terms of *(a,b)*. Thus we have a way of warping the panorama into an aerial (overhead) view (Fig. 3).

Fig. 3. (Left) The aerial view corresponding to the panorama shown in the previous figure. Notice that the two crosswalks can be seen to meet at approximately right angles in this view. (Right) Crosswalk features segmented from aerial view.

In our experiments a and b lie in the range $[-L, L]$, where $2L$ is the width of the aerial view region in meters. (L is typically set either to 5 or 10 in our experiments.) Any value of *(a,b)* that maps to a pixel location outside of the bounds of the panorama is set to intensity 0 in the aerial image. This explains the presence of a black, roughly circular region in each aerial view, which corresponds to the points near the user's feet that are too far below the horizon to be visible to the camera.

Note also that an aerial image only provides detail about a small portion of an entire traffic intersection, the area that lies close to where the user is standing. Moreover, the farther the points in the aerial image are from the origin, the more likely they are to be degraded by geometric distortion, which is caused by at least two sources of error: (a) error in the estimate of the Euler angles, which creates error in the estimation of pitch and yaw in the panorama; and (b) curvature in the road surface (which is a feature of most roads intended to facilitate water drainage), which is a violation of the ground plane assumption.

3.4. Creation of traffic intersection template

For each traffic intersection we created a template model as follows (Fig. 4). A Google Maps satellite image of the intersection was downloaded (at maximum zoom to provide the finest resolution); note that Google Maps specifies the scale of the imagery in meters per pixel. A simple Matlab script was written allowing the experimenters to click on the image to segment out the crosswalk features in it. We did not strive for an accurate representation of the crosswalk stripe widths (which are approximately 30 cm in the intersections we analyzed), and thus represented each stripe as a straight-line segment.

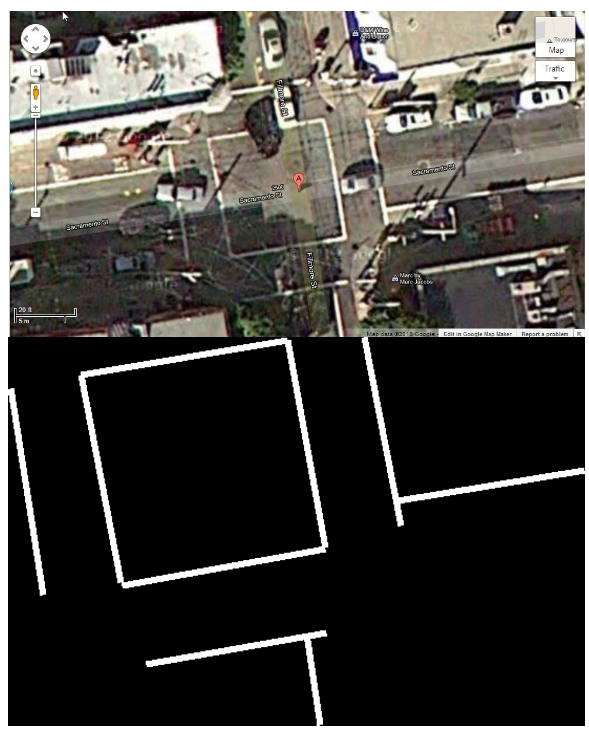

Fig. 4. (Top) Satellite image of intersection shown in previous figure. (Bottom) Template of the same intersection (zoomed in).

We note that in the future, templates could be created through a crowd-sourcing process [5] that recruits interested volunteers to make the appropriate annotations on Google

Maps imagery, which could be accessed as a browser-based tool. This crowd-sourcing process could add a variety of useful information to what is contained in Google Maps, including right-of-way information and even subjective opinions about which crosswalk markings are safer or easier to traverse than others. The use of crowd-sourcing could also alert Crosswatch users to those cases in which information in templates is out of date or otherwise incorrect, which may happen due to errors in Google Maps imagery or when it has not yet been updated to reflect recent changes in the actual crosswalk markings.

3.5. Matching the aerial view image to the template

Features in the aerial image have approximately the same scale and orientation as their corresponding matches in the template, which permits a direct comparison between them in the matching process. Since the aerial image only contains detail about the intersection near where the user is standing, the matching process searches for how the aerial view should be translated (in the *(x,y)* image plane) to best align with the template.

We perform several steps of image processing to segment the crosswalk features from the aerial image.

1.) First, the image is binarized by comparing the intensity of a pixel with the average intensity of the pixels in its neighborhood, to identify pixels that are brighter than their neighborhood (a similar procedure was used in [12]).

2.) Next, pixels corresponding to low intensity regions in the original image are removed from the binary image (crosswalk stripes have bright pixels).

3.) Border pixels of width 5 were set to intensity zero to prevent unwanted merging of pixels in the morphological operations (next step).

4.) Next, morphological operators are applied to remove small gaps in the crosswalk stripes.

5.) Connected components are extracted from the resulting binary map, with simple heuristics used to eliminate components that are very unlikely to belong to a crosswalk. These heuristics include the following:

   a) *Aspect ratio:* the ratio of the major axis length to the minor axis length of the connected component (in terms of the axes of the ellipse defined by the covariance matrix of the component). We discard those with too small an aspect ratio (i.e., components that are too "skinny").

   b) *Euler number:* a measure of the number of holes. Components with too many holes are discarded.

   c) *Area:* Components that are too small are discarded.

6.) Since the crosswalk scene observed by the user and its corresponding aerial image is a subset of the entire template of the intersection, it is necessary to search for the crosswalk scene in the satellite image to obtain the geographic position of the user with respect to the satellite image.

7.) The binary image obtained from steps 1-4 is matched to the template by searching for the image *(x,y)* translation that leads to the minimum matching cost. Here the matching cost is defined as the Hausdorff distance [13] between the two binary images. Rather than simply finding the value of *(x,y)* that minimizes the Hausdorff distance, a more robust procedure is used which finds the mean location about the pixels whose distance lies in the bottom 25% of distances.

Fig. 5 shows the resulting self-localization result from the matching procedure, which correctly identifies the user's location as being on the southeast corner of the intersection.

Fig. 5. Self-localization result from matching procedure shown as red asterisk superimposed on original satellite image. Result correctly identifies the user's location as being on the southeast corner of the intersection.

3.6. Experimental results

We show experimental results on two more image panoramas using imagery acquired by the two blind volunteers described above.

In some cases (e.g., Fig. 7), the orientation estimated in the aerial image deviates significantly from that of the template. While this problem needs to be addressed in the future (see next section), the matching is robust enough in some cases to arrive at a reasonable estimate of the user's location.

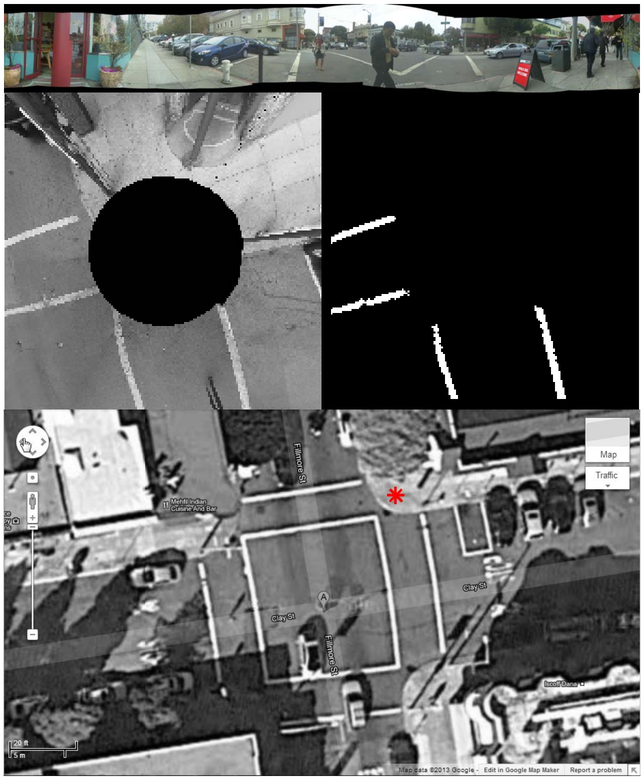

Fig. 6. More sample results. Top: image panorama. Middle: aerial view on left, crosswalk features segmented from aerial view on right. Bottom: self-localization result from matching procedure shown as red asterisk superimposed on original satellite image.

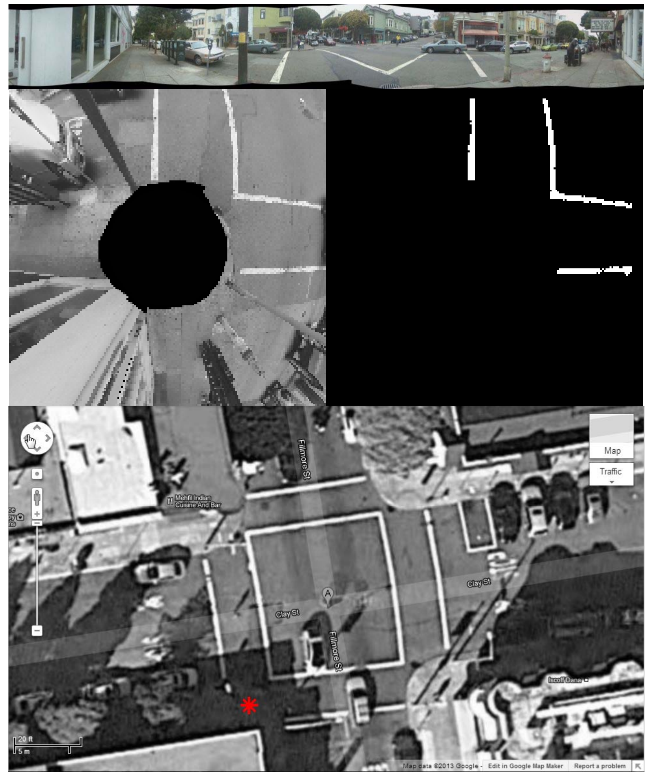

Fig. 7. More sample results. Top: image panorama. Middle: aerial view on left, crosswalk features segmented from aerial view on right. Bottom: self-localization result from matching procedure shown as red asterisk superimposed on original satellite image.

4. CONCLUSION AND DISCUSSION

We have described work in progress that tackles the problem of crosswalk detection and self-localization, using image panoramas acquired by blind or visually impaired users. The image panorama is converted to an aerial (overhead) view of the nearby intersection, centered on the location that the user is standing at, so as to facilitate matching with a template of the intersection obtained from Google Maps satellite imagery. The matching process allows crosswalk features to be detected and permits the estimation of the user's precise location relative to the crosswalk of interest. We demonstrate our approach on intersection imagery acquired by blind users, thereby establishing the feasibility of the approach.

Our current results are based on a preliminary set of algorithms that need further work and testing to be improved. First, the segmentation process needs to be made more robust, and needs to be extended to include other types of crosswalks, such as the "zebra" crosswalk, which has wide stripes. Currently we are not using color information in our algorithms, but most crosswalk stripes are either white or yellow, and since the color could be assumed known in the template this constraint could be used to improve the segmentation.

In addition, the matching process needs to be made more robust to the presence of outliers, missing points and geometric distortion. Moreover, while the process currently assumes the scales and orientations are identical in the aerial image and the template, this is not strictly the case, and some search over scale and orientation may be required to optimize the matching.

As mentioned previously, the creation of templates is somewhat time-consuming, and we will investigate methods for speeding up the process, and finding the relevant data in existing GIS (geographic information systems) and databases, or through the use of crowd-sourcing.

As an alternative to having users take image panoramas, which is a cumbersome process requiring them to stand in place and rotate in a full circle, we will also investigate the possibility of using wide-angle (e.g., fisheye) or catadioptric lenses (e.g., Kogeto: http://www.kogeto.com/). Such lenses could allow all the needed imagery to be acquired in a single video frame, but they have certain disadvantages as well, including the need for calibration, lower resolution relative to a full panorama, and the inconvenience of having to purchase and install an extra component on an otherwise off-the-shelf smartphone.

Even without the use of special lenses, in many cases a panorama far smaller than 360° (perhaps spanning 90-180°) should suffice for accurate localization results, and would be far more convenient than forcing users to stand in place and rotate in a full circle. This approach could be enabled by the use of the smartphone magnetometer, which would determine the approximate range of bearing needed to traverse a specific part of an intersection. (For instance, the user may know s/he wants to cross a specific street in an intersection in a specific direction, and the template of the intersection would determine the range of bearing needed accordingly.) A suitable user interface would then guide the user to sweep out the appropriate range of bearing to acquire the necessary imagery.

We will explore the possibility of augmenting the 2D image model of the intersection with 3D features from the environment. Compared with crosswalk features on the ground, such features may be less likely to be occluded by moving vehicles and pedestrians – and could even provide self-localization information at intersections that have no crosswalks at all.

Finally, we will need to improve the spatial resolution of our self-localization algorithms to permit detailed alignment information with respect to the crosswalk (as in [9]). To do so, we will need to obtain ground truth localizations to objectively evaluate the performance of our algorithms.

5. ACKNOWLEDGMENTS

The authors acknowledge support by the National Institutes of Health from grant No. 2 R01EY018345-05 and by the Department of Education, NIDRR grant number H133E110004. We would also like to thank Dr. Huiying Shen for his assistance with the image panorama acquisition app, which was originally developed in [8].

6. REFERENCES

[1] Ahmetovic, D., Bernareggi, C. and Mascetti, S. “Zebralocalizer: identification and localization of pedestrian crossings.” In Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11). ACM, New York, NY, USA. 2011.

[2] Aranda, J., Mares, P. “Visual System to Help Blind People to Cross the Street.” ICCHP 2004. Paris, France. LNCS 3118, pp. 454–461 (2004).

[3] Arth, C., Klopschitz, M., Reitmayr, G., & Schmalstieg, D. (2011, October). Real-time self-localization from panoramic images on mobile devices. In Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on (pp. 37-46). IEEE.

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

[5] Bigham, J. P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R. C., Tatrowicz, A., White, B., White, S. & Yeh, T. (2010, October). “VizWiz: nearly real-time answers to visual questions.” In Proceedings of the 23nd annual ACM symposium on User interface software and technology (pp. 333-342). ACM. 2010.

[6] Brabyn, J.A., Alden, A., Haegerstrom-Portnoy, G and Schneck, M. “GPS Performance for Blind Navigation in Urban Pedestrian Settings.” Proceedings, Vision 2002, Goteborg, Sweden, July 2002.

[7] Coughlan, J. and Shen, H. “The Crosswatch Traffic Intersection Analyzer: A Roadmap for the Future.” 13th International Conference on Computers Helping People with Special Needs (ICCHP '12). Linz, Austria. July 2012.

[8] Coughlan, J. and Shen, H. “Crosswatch: a System for Providing Guidance to Visually Impaired Travelers at Traffic Intersections.” Special Issue of the Journal of Assistive Technologies (JAT). Vol. 7, Issue 2. 2013.

[9] Ivanchenko, V., Coughlan, J. and Shen, H. “Staying in the Crosswalk: A System for Guiding Visually Impaired Pedestrians at Traffic Intersections.” Association for the Advancement of Assistive Technology in Europe (AAATE 2009). Florence, Italy. Sept. 2009.

[10] Ivanchenko, V., Coughlan, J. & Shen, H. “Real-Time Walk Light Detection with a Mobile Phone.” 12th International Conference on Computers Helping People with Special Needs (ICCHP '10). Vienna, Austria. July 2010.

[11] Ramalingam, S., Bouaziz, S., Sturm, P., & Brand, M. (2010, October). Skyline2gps: Localization in urban canyons using omni-skylines. In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on (pp. 3816-3823). IEEE.

[12] Sanketi, P., Shen, H. & Coughlan, J. “Localizing Blurry and Low-Resolution Text in Natural Images.” 2011 IEEE Workshop on Applications of Computer Vision (WACV 2011). Kona, Hawaii. Jan. 2011.

[13] Szeliski, R. *Computer vision: Algorithms and applications.* Springer, 2010.

[14] Uddin, M.S. and Shioyama, T. “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.

[15] Wiener, W.R., and Blasch, B.B. Foundations of orientation and mobility. Vol. 2. AFB Press, 2010.

[16] Zhu, Z., Oskiper, T., Samarasekera, S., Kumar, R., & Sawhney, H. S. (2008, June). “Real-time global localization with a pre-built visual landmark database.” In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on (pp. 1-8). IEEE.