

Computer Graphics – Literature Review 2

For this literature review, I chose to look at two papers that each propose a solution to the “Crosswatch” project, a theoretical computer vision program that can automatically find the distance and direction of a user from nearby crosswalks, with the primary goal of assisting visually impaired people in navigating urban environments. My primary essay is “Crosswalk Localization from Low Resolution Satellite Images to Assist Visually Impaired People”, written by Marcelo Ghilardi, Julio Jacques, and Isabel Manssour for the January 2018 edition of “IEEE Computer Graphics and Applications”. My secondary essay is “Smartphone-Based Crosswalk Detection and Localization for Visually Impaired Pedestrians”, written by Vidya Murali and James Coughlan for the July 2013 edition of “IEEE International Conference on Multimedia and Expo Workshops”.

To give more detail, the main problem both essays attempt to solve with regards to creating an effective “Crosswatch” program is how best to determine a user’s precise location when compared to nearby crosswalks whilst requiring as little input as possible from the user in order to function. To summarize briefly, both essays take a similar approach to solving this problem, opting to take advantage of a smartphone’s GPS coordinates to determine a user’s general location and comparing it with overhead pictures taken from the free Google Maps API to find the nearest crosswalks to the user. Where the two methods differ is in their priorities: my primary essay tries to make its solution as convenient for the user as possible at the cost of accuracy, relying almost entirely on GPS to determine a user’s position. Meanwhile, my secondary essay attempts to improve the accuracy of its results by comparing GPS results with those of other methods at the cost of user accessibility and development time.

To elaborate, my secondary essay, “Smartphone-Based Crosswalk Detection and Localization for Visually Impaired Pedestrians”, proposes a solution that determines a user’s location compared to crosswalks by comparing a panorama image taken by the user to both locational images downloaded from the Google Maps API as well as street templates manually created by the developer.

The method is as follows: When the user approaches an area where they believe an intersection is nearby, they use a specially-made application on their phone to take 18 pictures of their surrounding environment in a circle around them (with audio cues from the app informing the user when each picture is taken). These photos are then merged together to create a circular panoramic image which, using the user’s estimated height and the smartphone’s accelerometer and magnetometer as reference, is flattened into an aerial view of the surrounding area centered on the user.

From here, the aerial view produced is compared to a database containing multiple intersection templates. Each template represents a specific intersection in the writers’ home city of San Francisco and were drawn by a simple Matlab program that used aerial images of those intersections taken by Google Maps as a basis. The script takes a black-and-white image from Google Maps and create a new image that singles out the general shape and features of the crosswalk whilst ignoring all other surrounding area details. Once the application finds a

matching template, it then finds the position of the user by shifting the user's panorama image until it aligns with the template, making note of the distance in each direction it needed to move the image to complete the alignment. Finally, the application gives these distances to the user via audio cues. The essay also notes that, if this method were to be applied in an official release, the templates used for this experimental phase would be replaced with either a collection of templates generated via algorithm or by a more robust collection gathered via crowd-sourcing from willing users.

After testing, the writers of this essay found that their method was able to accurately determine a user's position relative to an intersection with a reasonable degree of accuracy, though there are quite a few downsides to the method as is. Firstly, it relies entirely on either a database existing that contains a template for the intersection being crossed (which, as stated above, would require either a more robust algorithm for template generation or a crowd-sourced solution) or the application itself creating new templates on the fly (which is incredibly time-consuming). Secondly, at the time of publication, the template generator isn't able to detect crosswalks whose lines are marked with non-white paint or crosswalks with a pattern other than the traditional "zebra" pattern (i.e. perpendicular white lines on black concrete). Lastly, while the method can accurately find the user's location, requiring the user to take a full panorama image is both time-consuming and taxing.

In response, my primary essay, "Crosswalk Localization from Low Resolution Satellite Images to Assist Visually Impaired People", proposes a solution that relies entirely on GPS coordinates to determine a user's location whilst skipping the template creation process by using an algorithm to gather needed information straight from Google Maps images in real-time.

Here's how the method works: To begin, the user turns on the developed application on their smartphone, which gathers the user's GPS coordinate data and sends it to an online web-service. This web-service then uses the Google Maps API to gather both the aerial view of the user's location and the road map of the surrounding area. From here, using the road map's measurements as a guideline, the webservice takes the aerial view obtained, greyscales it for the sake of image clarity, and segments out everything in the image except for the roads themselves.

The method's next step relies on an algorithm referred to as the Support Vector Machine classifier. This classifier is provided with 900 different 30 pixel by 30 pixel images, 370 of which being (and marked as) crosswalks and the other 530 being non-crosswalks. These images were then used as references to train the SVM in telling crosswalks from other objects in an image, with the images purposely varying in illumination, angle of orientation, and exact shape so as to improve the SVM's eventual effectiveness in different scenarios.

Going back to the method, the newly segmented image is divided into a series of 30 by 30 pixel squares that are then scanned over from top-to-bottom. If the SVM classifier recognizes any of these pixel sets as being part of a crosswalk, the method performs a second check on all surrounding pixel sets to see if any nearby also match. If there's another match during this check, then both patches are marked as crosswalks; If not, then the patch detected is ignored for the sake of accuracy. Once the scan is complete, the webservice estimates the user's distance from the nearest road and the directions needed for the user to reach the nearest crosswalk on that road, assuming that the user is facing towards said road. The generated directions are then passed back to the application, which provides them to the user via audio feedback.

After testing both their crosswalk detection method and user location method using 100 randomly selected crosswalks found on Google Maps, they found that both methods were incredibly effective, though with some limitations. Firstly, their SVM classifier had trouble detecting crosswalks whose symbols were either heavily obscured or damaged in some way, such as in the cities of developing countries like Brazil. Secondly, because of this application's reliance on GPS accuracy for determining a user's location, the application's effectiveness will go down in areas with less consistent GPS accuracy, such as less populated areas. However, even with these limitations, the writers found that their model as is was capable of detecting crosswalks with an accuracy of about 97% and detecting the locations of users relative to crosswalks with an accuracy of about 93% whilst only requiring an average of half a second to perform both estimations, which is incredibly effective for the time being.

Bibliography:

- 1) Crosswalk Localization from Low Resolution Satellite Images to Assist Visually Impaired People

@ARTICLE{7478448,
author={M. C. Ghilardi and J. Jacques Junior and I. Manssour},
journal={IEEE Computer Graphics and Applications},
title={Crosswalk Localization from Low Resolution Satellite Images to Assist Visually Impaired People},
year={2018},
volume={38},
number={1},
pages={30-46},
keywords={Computational modeling;Global Positioning System;Image processing;Image resolution;Object recognition;Rendering (computer graphics);Satellites;Image Based Modeling and Rendering;accessibility technologies;image processing;image segmentation},
doi={10.1109/MCG.2016.50},
ISSN={0272-1716},
month={Jan},}

- 2) Smartphone-based crosswalk detection and localization for visually impaired pedestrians

@INPROCEEDINGS{6618432,
author={V. N. Murali and J. M. Coughlan},
booktitle={2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)},
title={Smartphone-based crosswalk detection and localization for visually impaired pedestrians},
year={2013},
volume={},
number={},
pages={1-7},
keywords={computer vision;handicapped aids;object detection;pedestrians;smart phones;Crosswatch project;Google Maps satellite imagery;blind users;computer vision-based smartphone system;image panorama;intersection imagery;matching process;self-localization;smartphone-based crosswalk detection;smartphone-based crosswalk localization;traffic intersection;traffic intersections;user location estimation;visually impaired pedestrians;visually impaired travelers;Accelerometers;Cameras;Google;Magnetic levitation;Magnetic resonance imaging;Magnetometers;Satellites;Computer vision;blindness;self-localization;traffic intersection;visual impairment},
doi={10.1109/ICMEW.2013.6618432},
ISSN={},
month={July},}