Image Based Localization in Indoor Environments

Jason Zhi Liang
EECS Department
UC Berkeley
Berkeley, California
jasonzliang@eecs.berkeley.edu

Nicholas Corso
EECS Department
UC Berkeley
Berkeley, California
ncorso@eecs.berkeley.edu

Eric Turner
EECS Department
UC Berkeley
Berkeley, California
elturner@eecs.berkeley.edu

Avideh Zakhor
EECS Department
UC Berkeley
Berkeley, California
avz@eecs.berkeley.edu

***Abstract*—Image based localization is an important problem with many applications. In our previous work, we presented a two step pipeline for performing image based localization of mobile devices in outdoor environments. In the first step, a query image is matched against a georeferenced 3D image database to retrieve the "closest" image. In the second step, the pose of the query image is recovered with respect to the "closest" image using cell phone sensors. As such, a key ingredient of our outdoor image based localization is a 3D georeferenced image database. In this paper, we extend this approach to indoors by utilizing a 3D locally referenced image database generated by an ambulatory depth acquisition backpack that is originally developed for 3D modeling of indoor environments. We demonstrate retrieval rate of 94% over a set of 83 query images taken in an indoor shopping center and characterize pose recovery accuracy of the same set.**

***Keywords*—image retrieval, indoor localization, 3D reconstruction.**

I. Introduction

Indoor localization is an important problem with many useful applications such as geotagging and augmented reality. The most basic form of localization available to cell phones today is GPS. Unfortunately, GPS is only suitable for outdoor, rather than indoor environments. Furthermore, urban environments with tall buildings compromise GPS accuracy outdoors.

A number of alternative technologies have been proposed for indoor positioning systems over the years. These include optical [1], radio [2]–[6], magnetic [7], RFID [8], and acoustic [9]. Of these, most work has been focused on WiFi based localization which takes advantage of the proliferation of wireless access points. By using the user's cell phone to measure the signal strength of various wireless access points, the user's location is constrained to a relatively small area within a large indoor environment. However, there are several drawbacks to this approach. Aside from the fact that runtime operation of such a system requires the use of the same access points at the same location as in the calibration stage, its accuracy critically depends on the number of available access points. For instance, in order to achieve sub-meter accuracy, 10 or more wireless hotspots are typically required [3]. In indoor environments such as a mall, this would mean localization accuracy remains high in the center of the building, but drops sharply near the periphery, such as entrances or exits. Furthermore, such a system cannot determine the orientation of a user. This crucial shortcoming makes it impossible for WiFi localization by itself to support augmented reality applications.

Bluetooth beacons placed throughout the indoor environment can also be used for localization. Similar to WiFi, Bluetooth localization measures the user signal strength measured and is capable of achieving up to 1 meter level of accuracy [4]. While Bluetooth devices are relatively cheap and have high spatial selectivity, they experience high latency during the discovery phase [5].

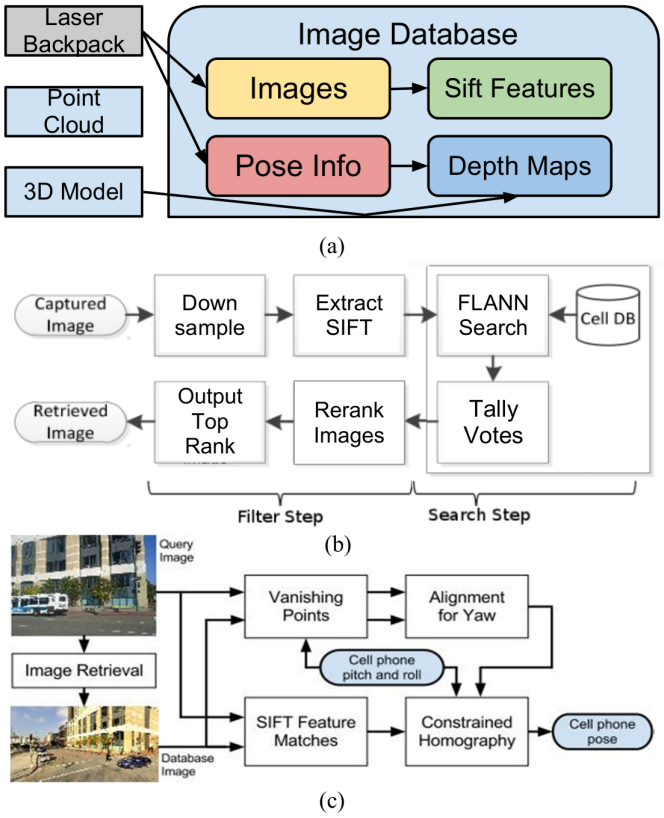

Fig. 1. Overview of our indoor localization pipeline. The pipeline is composed of (a) database preparation, (b) image retrieval, and (c) pose estimation stages.

There have also been previous attempts at indoor image based localization whereby information captured via the cell phone camera sensors are used to match images from a database [10]. The authors in [10] take advantage of off-the-shelf image matching algorithms, namely color histograms, wavelet decomposition, and shape matching and achieve room-level accuracy with more than 90% success probability, and meter-level accuracy with more than 80% success probability for one floor of the computer science building at Rutgers Uni-

versity. This approach however, cannot be used to determine the absolute metric position of the camera, nor its orientation. Thus it cannot be used in augmented reality applications where precise position and orientation is needed.

In this paper, we demonstrate an image based localization system for mobile devices which is not only capable of achieving sub-meter localization accuracy but also determines orientation. This system has the added advantage in that it requires no other hardware aside from the user's cell phone and a server to host an image database for its operation. Our proposed system consists of three components, shown in Fig. 1:

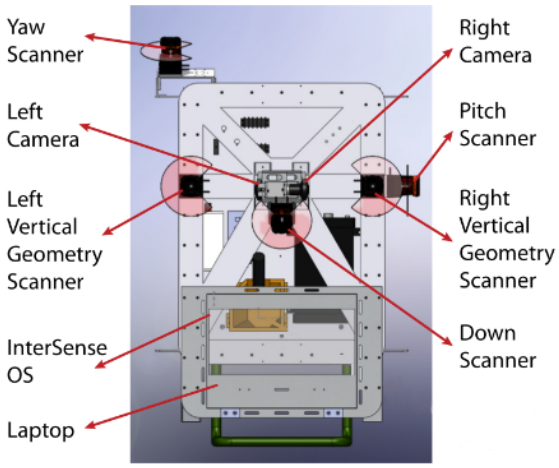

Fig. 2. Diagram of the data acquisition backpack.

(1) Database Preparation, shown in Fig. 1(a): We use a human operated ambulatory backpack, as seen in Fig. 2, outfitted with a variety of sensors to map the interior of a building in order to generate a locally referenced 3D image database complete with SIFT features [11]–[13]. By locally referenced image database, we mean that the absolute 6 degrees of freedom pose of all images, i.e. x, y, z, yaw, pitch, and row, are known with respect to a given coordinate system. By 3D, we mean there are depth values associated with each pixel in the database image.

(2) Image Retrieval, shown in Fig. 1(b): We load all of the image database SIFT features into a kd-tree and perform fast approximate nearest neighbor search to find a database image with most number of matching features to the query image [14]–[16].

(3) Pose Estimation, shown in Fig. 1(c): We use the SIFT feature matches along with cell phone pitch and roll to recover the relative pose between the retrieved database image in step (2) and the query image. This results in complete 6 degree of freedom pose for the query image in the given coordinate system [17].

The outline of the remainder of the paper is as follows: In Section II, we describe step (1) in more detail. In Section III, we examine steps (2) and (3) of our system. Section IV includes experimental setup and results.

II. Database Preparation

In order to prepare the image database, an ambulatory human operator first scans the interior of the building of interest using a backpack fitted with 2D laser scanners, fish-eye cameras, and inertial measurement units as shown in Fig. 2 [11]–[13]. Using scan matching algorithms, we localize the backpack over time by recovering its 6 degrees of freedom pose. Fig. 3(a) shows the recovered path of the backpack within the shopping center. Recovered pose and the rigidly mounted cameras on the backpack are used to generate a locally referenced image database in which the location, i.e. x, y, and z, as well as orientation, i.e. yaw, pitch, and roll, of each image is known.

Full recovery of the backpack pose also allows us to transform all the laser scans from the left and right vertical geometry scanners as well as the down scanner shown in Fig. 2 into a single 3D coordinate frame, which then results in a 3D point cloud as seen in Fig. 4(a) [12]. This point cloud and a novel surface reconstruction algorithm are used to reconstruct a 3D polygonal model of the building [18]. Fig. 4(b) shows an example of the model outputted by the algorithm. Given a registered point cloud representing indoor building geometry, this surface reconstruction algorithm generates a watertight triangulated surface that preserves sharp features and fine details of the building model. Since the mobile scanning system generates the point cloud, each point is produced with the scanner at a particular location and orientation. As such, the line-of-sight between the scanner and the resultant point is guaranteed to be free of obstacles. This surface reconstruction approach works by partitioning space into interior and exterior sets. The interior sets are "carved" away by labeling any volume that is intersected by such a scan-line as interior. Any space that is intersected by no such scan is considered exterior.

This labeling occurs on a voxel grid. By performing ray-tracing for each scan-line, all voxels are efficiently labeled as either interior or exterior. Once the voxel labeling is complete, the boundary between these two volumes are exported as a watertight surface. Ensuring a watertight surface is crucial, so that a depth map of the environment is recoverable from any orientation. This boundary surface is segmented into planar regions, which are then adaptively triangulated as seen in Fig. 4(b). This piece-wise planar segmentation yields three advantages. First, it matches prior knowledge of building interiors, which often consist of flat regions intersecting at sharp corners, such as floors, walls, and ceilings. Second, the planarity assumption ensures that the plane equation is known for each surface of the model, thereby simplifying the homography computation for the pose recovery step. Third, this segmentation allows for efficient triangulation of the surface, yielding faster raytracing performance and lower memory overhead during processing. The output surface is one that preserves enough detail to depict the important features for indoor pose recovery, such as the geometry of store-fronts, while still providing a simple enough model to allow for efficient processing.

With a simplified 3D model of the building interior, the task of generating depthmaps for each image becomes relatively straightforward. Making use of each database image's location, pose, and intrinsic parameters, we trace a ray starting from the camera origin of the database image and through each pixel

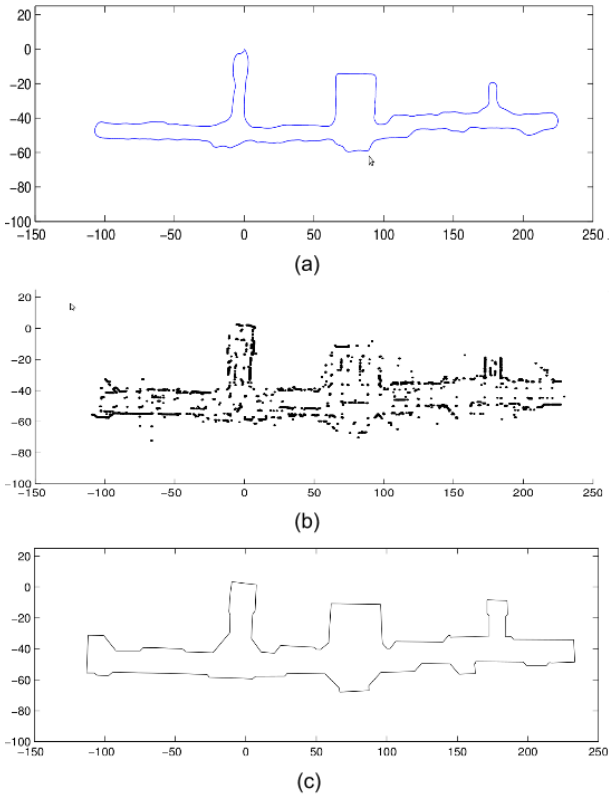

Fig. 3. (a) Recovered path of backpack traversal. (b) Wall points generated by backpack. (c) 2D floorplan recovered from wall points.

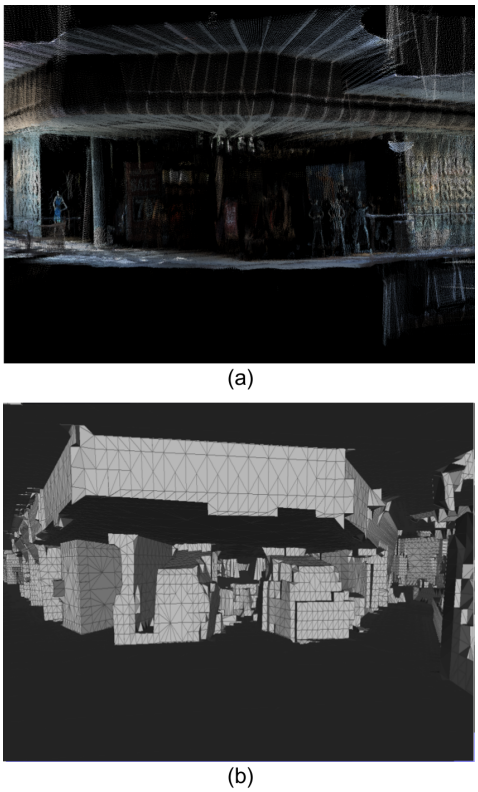

Fig. 4. (a) Point cloud generated by backpack. (b) 3D model reconstructed from point cloud.

of the image. The ray is traced until it intersects with the first piece of geometry in the path of the ray. In doing so, we are able to calculate the distance from the camera origin to the intersecting location and determine a depth value for every pixel in the image. The water tightness of the 3D model guarantees successful intersection for a ray traced from any arbitrary direction.

In order to reduce intersection tests and the time needed to generate depthmaps, we implement two optimizations. The first one is a kd-tree acceleration structure that drastically reduces the amount of geometry needed for intersection tests. By dividing the 3D model into axis aligned bounding boxes, intersection tests are only performed for polygons located in the subset of bounding boxes that the ray travels through. The second one is to cache polygons that have been recently intersected since these are likely to be intersected again later in the raytracing. Fig. 5(b) shows the 3D model when viewed from the database image's pose while Fig. 5(c) shows an raytraced depthmap of the same image.

The 3D point cloud resulting from the backpack are also used to create an approximate floorplan for the building of interest. The procedure is as follows: As shown in Fig. 3(b), we find a sparse sampling of points that represent wall locations using the 2D histogram approach described in [19]. These samples are essentially projections of geometry scanners onto the 2D horizontal ground plane. As shown in Fig. 3(c), the samples are then connected interactively using CAD software to produce a set of lines which approximate the true layout of the walls. As a last step, we extract SIFT features from every database image for later use along the image localization pipeline [15].

III. Image Retrieval and Pose Estimation

The next step of our image localization pipeline shown in Fig. 1(b) is image retrieval, which involves selecting the best matching image from the image database for a particular query image. Our indoor image retrieval system loads the SIFT features of every database image into a single FLANN kd-tree [16]. Next, we extract SIFT features from the query image and for each SIFT vector extracted, we lookup its top N neighbors in the kd-tree. For each closest neighbor found, we assign a vote to the database image that the closest neighbor feature vector belongs to. Having repeated this for all the SIFT features in the query image, the database images are ranked by the number of matching SIFT features they share with the query image. We find that a value of 4 for N results in optimal image retrieval performance.

After tallying the votes, we perform geometric consistency checks and rerank the scores to filter out mismatched SIFT features. We solve for the fundamental matrix between the database and query images and eliminate feature matches which do not satisfy epipolar constraints [14]. We also remove SIFT feature matches where the angle of SIFT features differ by more than 0.2 radians. Because these geometric consistency checks only eliminate feature matches and decrease the scores of database images, we only need to partially rerank the database images. The database image with the highest score

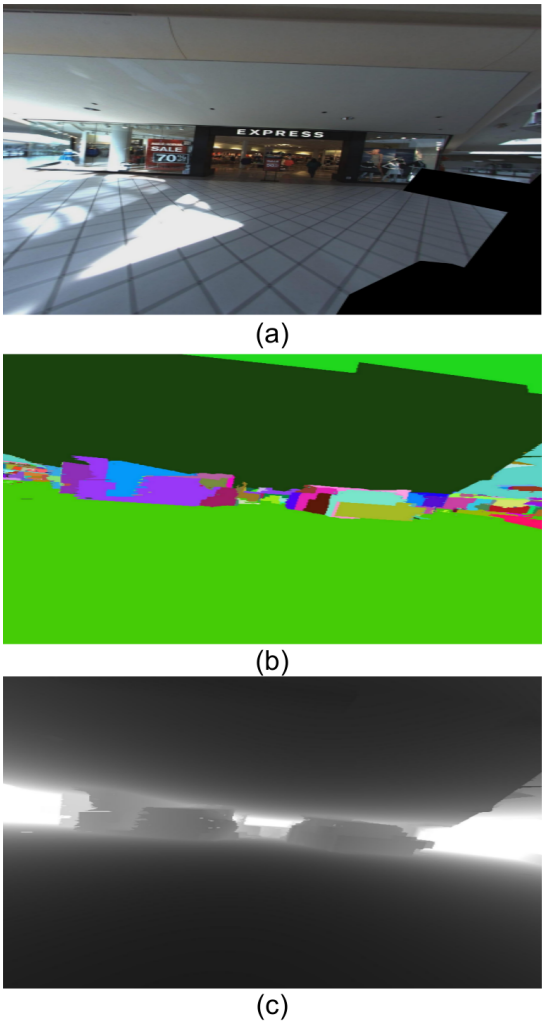

Fig. 5. (a) Original database image. (b) 3D model viewed from pose of database image. (c) Raytraced depthmap of database image.

after reranking is outputted as the best match to the query image.

As shown in Fig. 1(c), the last step of our indoor localization pipeline is pose recovery of the query image [17]. Pitch and roll estimation from cell phone sensors are used in vanishing point analysis to compute yaw of the query image. Once we estimate complete orientation, SIFT matches are used to solve a constrained homography problem to recover translation between query and database images.

IV. Experimental Results

For our experimental setup, we use the ambulatory human operated backpack of Fig. 2 to scan the interior of Newpark Mall, a two story shopping center at Fremont, California. To generate a 3D locally referenced image database, we collect over 20,000 images with the fish-eye cameras mounted on the backpack. These 20,000 images are then rectified into rectilinear images to remove the fish-eye distortion. Since the images overlap heavily with each other, it is sufficient to include every sixth image for use in the database. By reducing the number of images, we are able to speed up image retrieval by several factors with virtually no loss in accuracy. The 3D model resulting from the surface reconstruction algorithm in [18] has a visual resolution of 30 cm. Raytracing this model generates depth values for all pixels in the image database. We have found that a relatively coarse resolution is sufficient for the pose estimation portion of the pipeline.

Our query image data set consists of 83 images taken with a Samsung Galaxy S3 smartphone. The images are approximately 5 megapixels in size and are taken using the default settings of the Android camera application. Furthermore, the images consist of landscape photos either taken head-on in front of a store or at a slanted angle of approximately 30 degrees. After downsizing the query images to the same resolution as the database images, i.e. 1.25 megapixels, we successfully match 78 out of 83 images to achieve a retrieval rate of 94%. After detailed analysis of the failure cases, we have found that two of the incorrectly matched query images are of a store that does not exist in the image database. This means the true failure rate of our image retrieval system is 3 out of 80 or less than 4%. As shown in Fig. 6(a), successful retrieval usually involves matching of store signs present in both the query and database images. In cases where retrieval fails, i.e. Fig. 6(b), there are few features on the query image's store sign that get matched.

Next, we run the remaining query images with successful retrieved database images through the pose estimation part of the pipeline. In order to characterize pose estimation accuracy, we first manually ground truth the position and pose of each query image taken. This is done by using the 3D model representation of the mall and distance measurements recorded during the query dataset collection. For each query image, we are able to specify a ground truth yaw and position in the same coordinate frame as the 3D model and the output of the pose recovery step.

Fig. 7 summarizes the performance of the pose estimation stage of our pipeline. As shown in Figs. 7(a) and 7(c), we are able to localize the position to within sub-meter level of accuracy for over 50% of the query images. Furthermore, 85% of the query images are successfully localized to within two meters of the ground truth position. As shown in Figs. 7(b) and 7(d), we are able to correctly estimate yaw within 10 degrees of ground truth for roughly 90% of the query images. As seen in Fig. 8(a), when the location error is less than 1 meter, the SIFT features of corresponding store signs present in both query and database images are matched together by the RANSAC homography [17]. Conversely, in less accurate cases of pose estimation where the location error exceeds 4 meters, the RANSAC homography finds "false matches" between unrelated elements of the query and database images. In the example shown by Fig. 8(b), different letters in the signs of the two images are matched together. In general we find that images with visually unique signs perform better during pose estimation than those lacking such features. In Fig. 9, we plot the estimated and ground truth locations of the query images onto the shopping center's 2D floorplan. As seen from this figure, there is close agreement between the two.

On a 2.3 GHz i5 laptop, our complete pipeline from image retrieval to pose recovery takes on average 10-12 seconds to output a solution for a single image. On an Amazon EC2 extra-large computing instance, the runtime is reduced further to an

Fig. 6. (a) Successful and (b) unsuccessful examples of image retrieval. Red lines show SIFT feature matches.

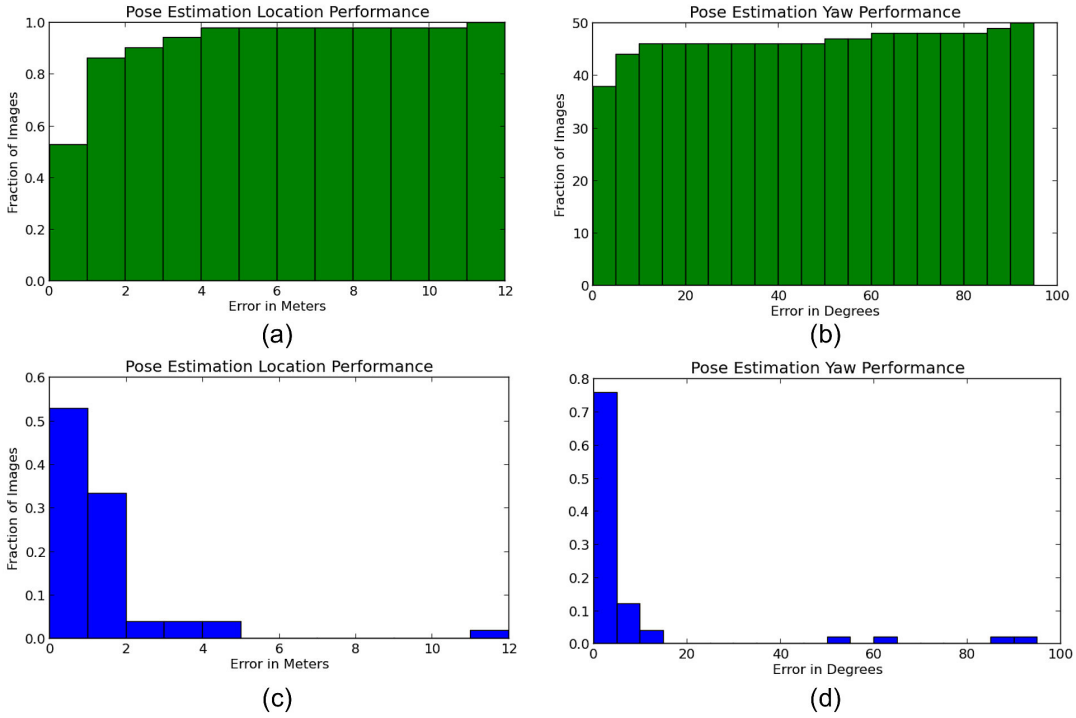

Fig. 7. Cumulative histogram of (a) location, (b) yaw error. Probability distribution function of (c) location, (d) yaw error.

average of 4.5 seconds per image.

V. Conclusion

In this paper, we have presented a complete image based localization pipeline for indoor environments. Our system is capable of achieving accuracy matching or exceeding that of WiFi based localization systems and can determine the pose of the user as well. For future work, we plan to explore ways to further optimize the speed and accuracy of our pipeline, integrate an online tracking algorithm, and replace SIFT with image descriptors that are more robust to viewpoint variation.

References

[1] L. I. U.Xiaohan, Hideo Makino, and M. A. S. E. Kenichi, "Improved Indoor Location Estimation using Fluorescent Light Communication System with a Nine-channel Receiver," in IEICE Transactions on Communications 93, no. 11 (2010): 2936-2944.

[2] Yongguang Chen and Hisashi Kobayashi, "Signal Strength Based Indoor Geolocation," in International Conference on Communications, 2002.

[3] Joydeep Biswas and Manuela Veloso, "WiFi Localization and Navigation for Autonomous Indoor Mobile Robots," in International Conference on Robotics and Automation, 2010.

[4] Gunter Fischer, Burkhart Dietrich, and Frank Winkler, "Bluetooth Indoor Localization System," in Proceedings of the 1st Workshop on Positioning, Navigation and Communication, 2004.

[5] Sudarshan S. Chawathe, "Low-latency Indoor Localization using Bluetooth Beacons," in 12th International IEEE Conference on Intelligent Transportation Systems, 2009.

(a)

(b)

Fig. 8. (a) Example of accurate pose estimation on query image. (b) Example of inaccurate pose estimation. Notice how different letters in the same sign are matched.

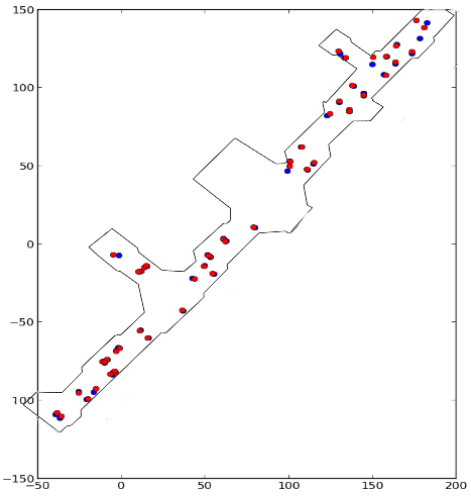

Fig. 9. A plot of ground truth location (red) and computed location (blue) of query images onto a 2D floorplan of the mall.

[6] Alex Varshavsky, Eyal de Lara, Jeffrey Hightower, Anthony LaMarca, and Veljo Otsason, "GSM Indoor Localization," in Pervasive and Mobile Computing 3, no. 6 (2007): 698-720.

[7] Jaewoo Chung, Matt Donahoe, Chris Schmandt, Ig-Jae Kim, Pedram Razavai, and Micaela Wiseman, "Indoor Location Sensing using Geo-magnetism," in Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, pp. 141-154. ACM, 2011.

[8] Sebastian Schneegans, Philipp Vorst, and Andreas Zell, "Using RFID Snapshots for Mobile Robot Self-Localization," in European Conference on Mobile Robots, 2007.

[9] Hong-Shik Kim, and Jong-Suk Choi, "Advanced Indoor Localization using Ultrasonic Sensor and Digital Compass," in International Conference on Control, Automation and Systems, 2008.

[10] Nishkam Ravi, Pravin Shankar, Andrew Frankel, Ahmed Elgammal, and Liviu Iftode, "Indoor Localization using Camera Phones," in Mobile Computing Systems and Applications, 2006.

[11] G. Chen, J. Kua, S. Shum, N. Naikal, M. Carlberg, and A. Zakhor. "Indoor Localization Algorithms for a Human-Operated Backpack System," in 3D Data Processing, Visualization, and Transmission, May 2010.

[12] T. Liu, M. Carlberg, G. Chen, Jacky Chen, J. Kua, and A. Zakhor, "Indoor Localization and Visualization Using a Human-Operated Backpack System," in International Conference on Indoor Positioning and Indoor Navigation, 2010.

[13] J. Kua, N. Corso, A. Zakhor, "Automatic Loop Closure Detection Using Multiple Cameras for 3D Indoor Localization," in IS&T/SPIE Electronic Imaging, 2012.

[14] Jerry Zhang, Aaron Hallquist, Eric Liang, and Avideh Zakhor, "Location-Based Image Retrieval for Urban Environments," in International Conference on Image Processing, 2011.

[15] D. G. Lowe, "Distinctive Image Features from Scale- Invariant Keypoints," in International Journal of Computer Vision, 60, 2 (Jan. 2004), 91-110.

[16] M. Muja and D. G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration," in International Conference on Computer Vision Theory and Applications, 2009.

[17] Aaron Hallquist and Avideh Zakhor, "Single View Pose Estimation of Mobile Devices in Urban Environments," in Workshop on the Applications of Computer Vision, 2013.

[18] Eric Turner and Avideh Zakhor, "Watertight Planar Surface Meshing of Indoor point clouds with Voxel Carving," unpublished paper in 3D Vision, 2013.

[19] Eric Turner and Avideh Zakhor, "Watertight As-Built Architectural Floor Plans Generated From Laser Range Data," in 3D Imaging, Modeling, Processing, Visualization and Transmission, 2012.

[20] Link to 3D Model: http://www-video.eecs.berkeley.edu/research/indoor/mmsp2013_mall_res_30_cm.ply