Reduced-Complexity Data Acquisition System for 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 has important commercial applications such as augmented reality and customer analytics. In prior work, we developed a three step pipeline for image-based localization of mobile devices in indoor environments. In the first step, we generate a 2.5D georeferenced image database using an ambulatory backpack-mounted system originally developed for 3D modeling of indoor environments. Specifically, we first create a dense 3D point cloud and polygonal model from the side laser scanner measurements of the backpack, and then use it to generate dense 2.5D database image depthmaps by raytracing the 3D model. In the second step, a query image is matched against the image database to retrieve the best-matching database image. In the final step, the pose of the query image is recovered with respect to the best-matching image. Since the pose recovery in step three only requires sparse depth information at certain SIFT feature keypoints in the database image, in this paper we improve upon our previous method by only calculating depth values at these keypoints, thereby reducing the required number of sensors in our data acquisition system. To do so, we use a modified version of the classic multi-camera 3D scene reconstruction algorithm, thereby eliminating the need for expensive geometry laser range scanners. Our experimental results in a shopping mall indicate that the proposed reduced complexity sparse depthmap approach is nearly as accurate as our previous dense depth map method.**

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

I. Introduction

Indoor localization allows for many commercially viable applications, such as customer navigation, behavior and movement tracking, and augmented reality (AR). These applications all require the user's location and orientation to be reliably estimated. Localization is noticeably more challenging indoors than outdoors since GPS is typically unavailable in interior environments due to the shielding effect of structures. As a result, much research has been focused on relying on other types of signals, or in our case, images as a basis for localization.

A variety of sensors are capable of performing indoor localization, including image [1], optical [2], radio [3]–[7], magnetic [8], RFID [9], and acoustic [10]. WiFi based indoor localization takes advantage of the proliferation of wireless access points (AP) and WiFi capable smartphones and uses the signal strength of nearby WiFi beacons to estimate the user's location. A few drawbacks are that APs cannot be moved or modified after the initial calibration, and that a large of number of APs are required to achieve reasonable accuracy. For instance, 10 or more wireless hotspots are typically required to achieve sub-meter accuracy [4]. The most debilitating drawback of WiFi localization is its inability to estimate the user's orientation, which is necessary for AR applications. Other forms of indoor localization that rely on measuring radio signal strength such as bluetooth, GSM, and RFID, also share the same strengths and weaknesses of WiFi based indoor localization.

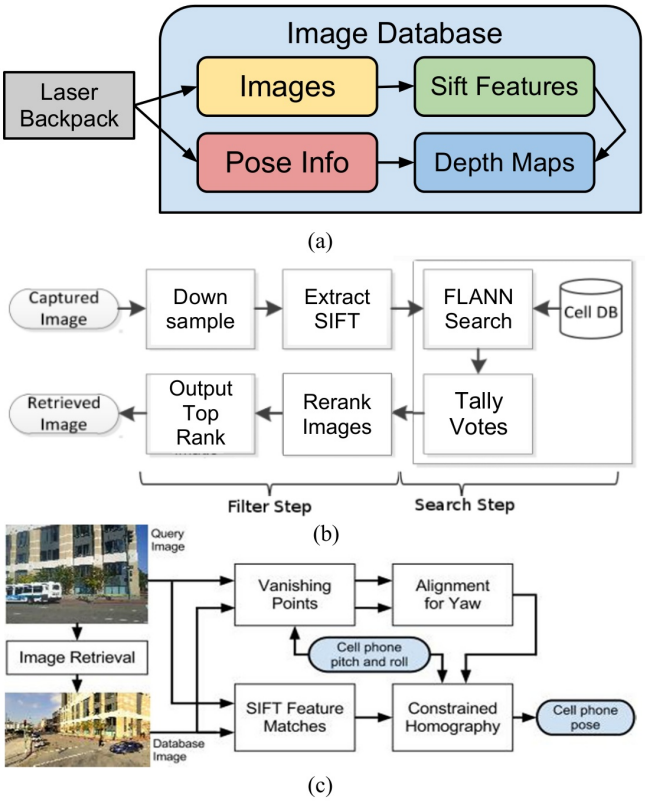


Fig. 1. Overview of our indoor localization pipeline. The pipeline is composed of (a) database preparation, (b) image retrieval, and (c) pose estimation stages.

In this paper, we take advantage of another sensor readily available on modern smartphones for image localization: images taken by the camera. An image-based localization system involves retrieving the best image from a database that matches

to the user's query image, then performing pose estimation on the query/database image pair in order to estimate the location and orientation of the query image. Previous attempts to image based localization involve position recovery only and typically do not estimate orientation [11].

In prior work, we demonstrated an image-based localization system for mobile devices capable of achieving sub-meter position accuracy as well as orientation recovery [1]. The three stages of that pipeline are: (1) preparing a 2.5D locally referenced image database, (2) image retrieval, and (3) pose recovery from the retrieved database image. To generate a dense 3D point cloud to be used in step 1 in [1], we use the ambulatory backpack mounted system shown in Fig. 2 which consists of five 2D laser range sensors (LRS), two cameras, and one orientation sensor (OS). In this paper, we show that a reduced complexity data acquisition system consisting of two LRSs, one camera, and one OS is sufficient to achieve nearly identical localization accuracy as the more complex system in [1]. Such a reduced complexity system is not only significantly less expensive from a hardware viewpoint, but also is simpler from a computational point of view in that it obviates the need for 3D point cloud and 3D model generation [12]. The main idea behind the reduced complexity approach in this paper is that the pose recovery in step 3 of [1] only requires sparse depth information of SIFT feature keypoints in the database image, rather than the dense depth at every pixel. As such, it is possible to use a modified version of the classic multi-camera 3D reconstruction algorithm [13] by taking advantage of the fact that relative pose of the camera can be obtained via 2D localization algorithms based on the yaw scanner [14]–[16].

We also present a method to estimate confidence values for both image retrieval and pose estimation of our proposed image-based localization system. These two confidence values can be combined to form an overall confidence indicator. Furthermore, the confidence values for our pipeline can be combined with that of other sensors such as WiFi in order to yield a more accurate result than each method by itself.

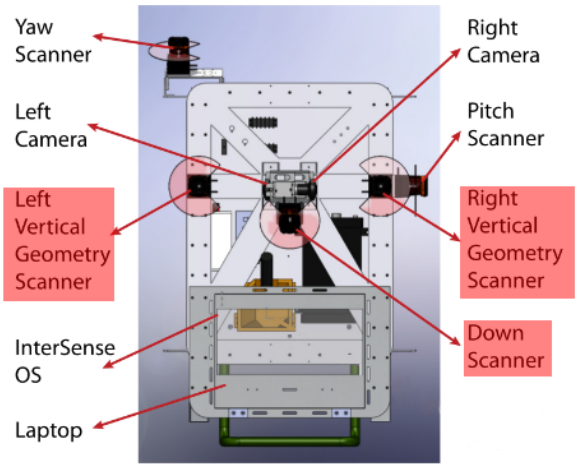


Fig. 2. Diagram of the data acquisition backpack. In our new pipeline, the left, right, and down scanners, highlighted in red, are no longer needed.

Our new pipeline can be summarized as follows:

(1) Database Preparation, shown in Fig. 1(a): We use a human operated ambulatory backpack outfitted with a yaw scanner, a pitch scanner, two cameras, and an OS, as seen in Fig. 2, to map the interior of a building in order to generate a locally referenced sparse 2.5D image database complete with SIFT features [14]–[16]. 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 sparse 2.5D, we mean there are depth values associated with SIFT feature keypoints in each 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 [17]–[19].

(3) Pose Estimation, shown in Fig. 1(c): We use the depth of 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 [20].

In Section II, we describe the database preparation setup. In Section III, we will go over image retrieval and pose estimation. Section IV includes estimation of confidence values for both image retrieval and pose estimation are estimated. In Section V, we show experimental results, comparing the accuracy of our new pipeline to the old one [1]. In Section V includes conclusions and future work.

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 [14]–[16]. Unlike our previous approach which requires five laser range scanners shown in Fig. 2, the database acquisition system in this paper only requires two laser scanners, namely the pitch and yaw scanners in Fig. 2. Measurements from the backpack's yaw and pitch laser range scanners are processed by a scan matching algorithm to localize the backpack at each time-step and recover its 6 degrees of freedom pose. Specially, the yaw scanner is used in conjunction with a 2D localization algorithm in [14]–[16] to recover x, y and yaw, the OS is used to recover pitch and roll, and the pitch scanner is used to recover z [14]. Since the cameras are rigidly mounted on the backpack, recovering the backpack pose essentially implies recovering camera pose. Fig. 3(a) shows the recovered path of the backpack within a shopping center, while Fig. 3(b) shows the surrounding wall points recovered by the backpack by projecting the yaw scans onto the XY plane [21]. These wall points can be connected interactively via commercially available CAD software to produce an approximate 2D floor-plan of the mall as seen in Fig. 3(c). The recovered pose of the rigidly mounted cameras on the backpack are then 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.

To create a sparse depthmap for the database images, we first temporally sub-sample successive captured images

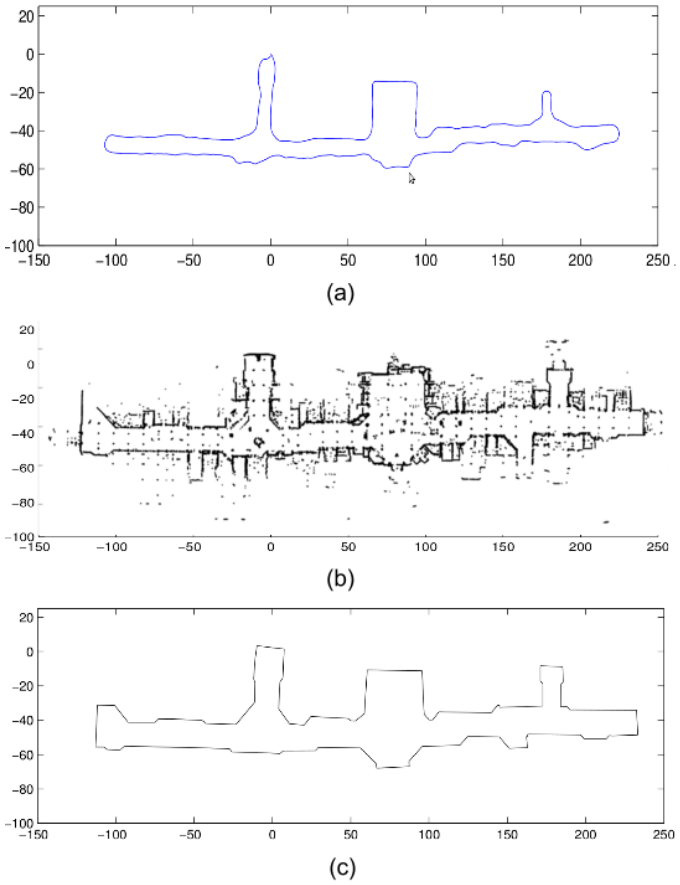


Fig. 3. (a) Recovered path of backpack traversal. (b) Wall points generated by backpack. (c) 2D floorplan recovered from wall points.

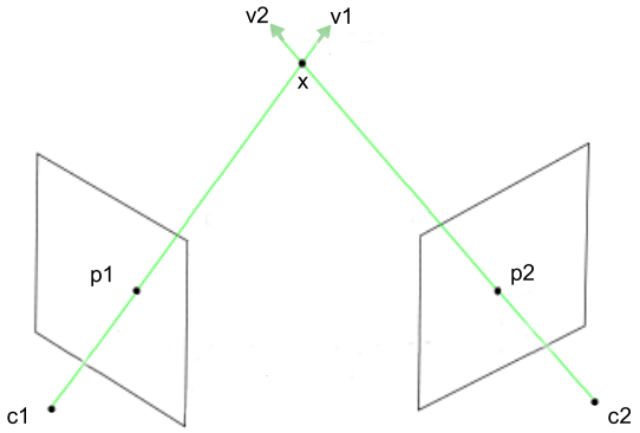


Fig. 4. Triangulation of two matching SIFT features. v1 and v2 are the resulting vectors when the camera centers c1 and c2 are connected to the SIFT features p1 and p2 on the image planes. The two vectors intersect at x.

on the backpack while maintaining their overlap. We then extract SIFT features from each pair of images and determine matching feature correspondence pairs through nearest-neighbor search. In order to ensure the geometric consistency of the SIFT features, we compute the fundamental matrix that relates the two sets of SIFT features and removes any feature pairs which do not satisfy epipolar constraints.

We then triangulate matching SIFT keypoint pairs in 3D space. As seen in Fig. 4, for each pair of SIFT correspondences, we calculate the 3D vectors that connects the camera centers of the images to the respective pixel locations of their SIFT features. In doing so, we make use of the database images' poses and intrinsic parameters to ensure both vectors are correctly positioned within the same world coordinate frame. Next, we determine the depth of the SIFT features by finding the intersection of these rays and computing the distance from camera center to the intersection point. We use the following to determine the intersection point or the point mutually closest to the two vectors:

$$x = \left(\sum_{i=1}^{2} I - v_i v_i^T \right)^{-1} \left(\sum_{i=1}^{2} \left(I - v_i v_i^T \right) p_i \right) \qquad (1)$$

where x is the intersection point, v_i is the normalized direction of the i^{th} vector, and p_i is a point located on the i^{th} vector. The availability of highly optimized library functions for determining fundamental matrices and performing linear algebra operations means that sparse depthmap generation can be done in a matter of seconds per image. As such, the runtime for sparse depthmap generation is an order of magnitude smaller than that for our previous method of raytracing dense depthmaps [1]. For debugging and visualization purposes, we combine the intersection points of SIFT features from every database image into a single sparse 3D point cloud, shown in Figs. 5(a) and (b). In comparison, the dense 3D point clouds generated in our previous work [1] are shown in Figs. 5(c) and (d).

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 [19]. 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.

After tallying the votes, we check geometric consistency and rerank the scores to filter out mismatched SIFT features. We then solve for the fundamental matrix between the database and query images and eliminate feature matches that do not satisfy epipolar constraints [17]. We also remove SIFT feature matches where the angle of SIFT features differ by more than 0.2 radians. Since 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 after reranking is exported as the best match to the query image. The image retrieval step takes roughly 2-4 seconds depending on the processing power of the CPU used.

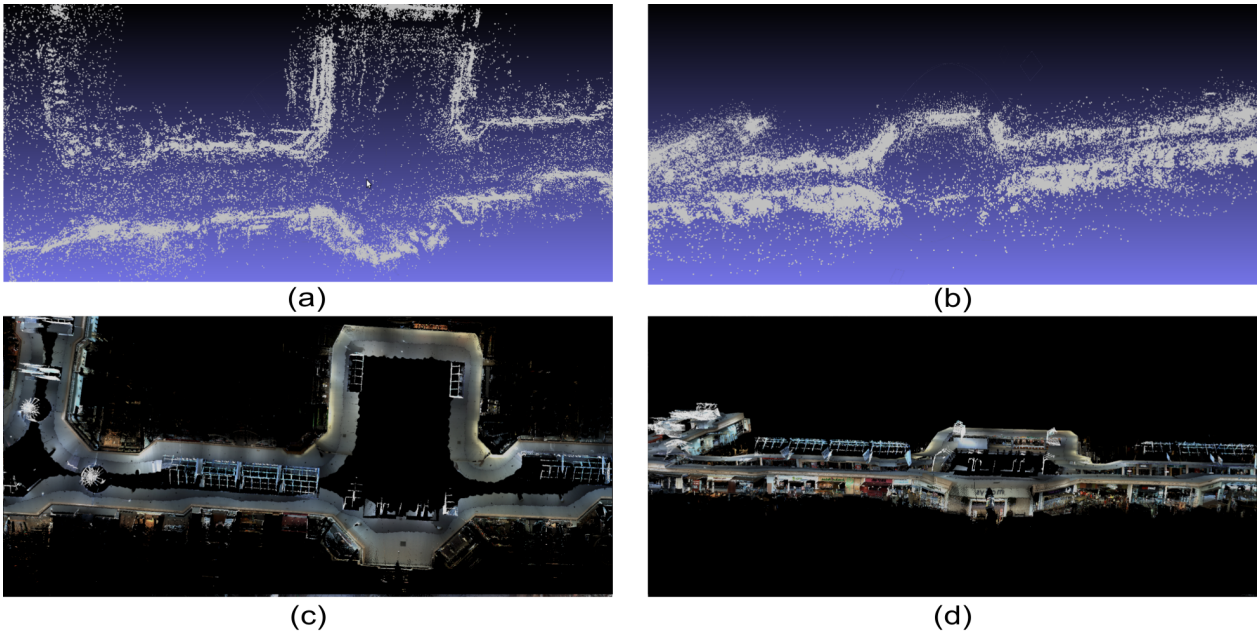


Fig. 5. (a) Top-down and (b) side views of sparse 3D point cloud generated from triangulation of SIFT feature correspondence of the database images; (c) and (d) are same views of (a) and (b) but for the dense point cloud generated in previous work [1]

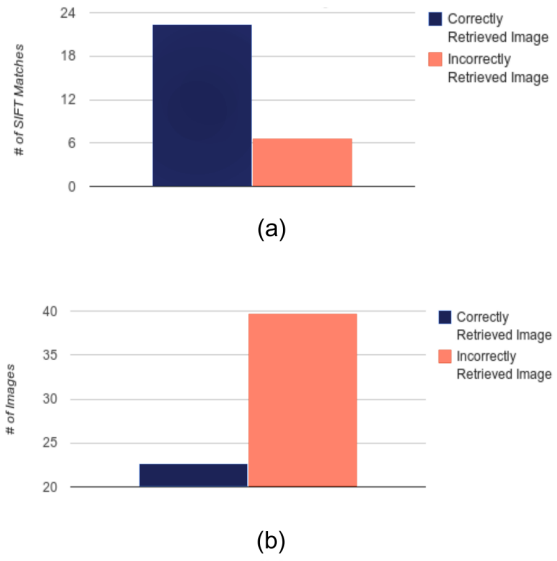


Fig. 6. Comparison of (a) number of SIFT matches after geometric consistency check and (b) vote ranking distribution before geometric consistency check for correctly and incorrectly retrieved images.

As shown in Fig. 1(c), the last step of our indoor localization pipeline is pose recovery of the query image. Pitch and roll estimates from cell phone sensors are used in vanishing point analysis to compute yaw of the query image [20]. Once we estimate orientation, SIFT matches are used to solve a constrained homography problem within RANSAC to recover translation between query and database images. The method for scale recovery of the translation vector only requires depth values at the SIFT features which are considered inliers from the RANSAC homography. These sparse depth values are generated earlier during the database preparation of section II. Occasionally, when the inlier has no corresponding depth value, we look up the depth value of the closest neighboring SIFT feature in the depthmap. We have also found that reducing the size of the query images significantly reduces the number of iterations required for RANSAC homography. This is because the resolution of our database images is significantly lower than that of the query image camera. If the RANSAC homography fails to find inliers, we use the pose of the matched database image as the solution. Depending on the image and speed of the CPU, pose estimation requires 2-10 seconds.

IV. Confidence Estimation

Our confidence estimation system consists of several classifiers that output confidence values for both the image retrieval and pose recovery steps in our pipeline. These classifiers are trained using positive and negative examples from both image retrieval and pose recovery stages of our proposed pipeline in Section III. We have experimentally found a logistic regression classifier to perform reasonably well even though other classifiers can also be used for confidence estimation. In order to evaluate the performance of our confidence estimation system, we create a dataset of over 270 groundtruth images where roughly 25% of the images are used for validation and the rest for training. To boost classifier performance, 50 out of the 270 images in the validation set are chosen to be "negative" images that do not match to any image database.

To generate confidence values for image retrieval, we train a logistic regression classifier based on features obtained during the image retrieval process. We assign groundtruth binary labels to the images in the training set that indicate

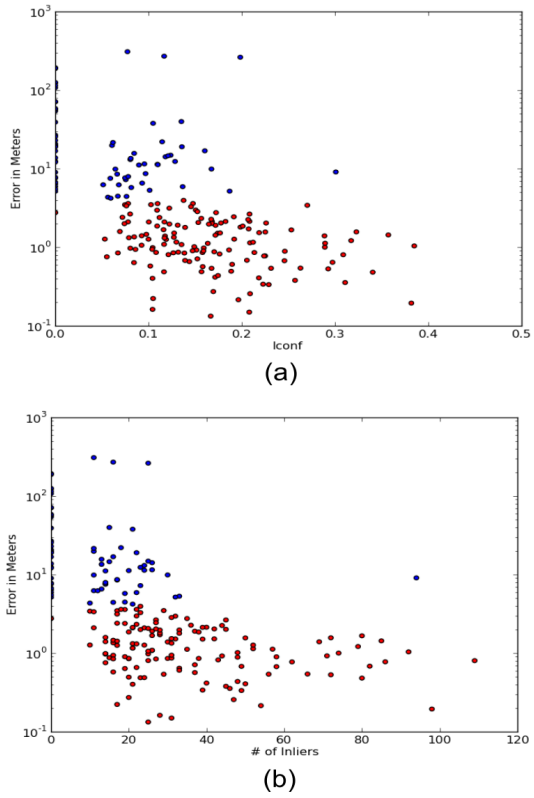


Fig. 7. Scatterplot of (a) confidence metric used in [20] and (b) number of inliers after RANSAC homography versus location error. Red (blue) dots correspond to images with less (more) than 4 meters of location error.

whether the retrieved images matches the query images. For a given query image, the retrieval classifier generates both a predicted binary label and a retrieval confidence value between 0 and 1. We have found the following features to be well correlated with image retrieval accuracy [17]: (a) number of SIFT feature matches between query and database image before geometric consistency checks; (b) number of SIFT matches after geometric consistency checks; (c) the distribution of the vote ranking before geometric consistency checks; (d) the distribution of the vote ranking after geometric consistency checks; (e) physical proximity of the top N database images in the vote ranking. For example, as shown in Fig. 6(a), the average number of SIFT matches after geometric consistency checks for correctly matched query images is over 3 times that of incorrectly matched query images. Likewise, as shown in Fig. 6(b), the number of database images with at least half the number of votes of the top ranked image before the geometric consistency check is much lower for correctly retrieved images than the incorrectly retrieved ones.

Similarly for pose estimation, we train a separate logistic regression classifier on another set of features that correlate well with the pose recovery accuracy. We assign a groundtruth "True" label if the location error of a training image is below a pre-specified threshold t_s = 4 meters, and a "False" label otherwise. As with the image retrieval classifier, our pose estimation classifier generates a predicted binary label and a confidence value between 0 and 1. The features use to train the classifier are: (a) number of inliers after RANSAC homography; (b) reprojection error; (c) number of SIFT feature matches before RANSAC homography; (d) number of RANSAC iterations; (e) a confidence metric in [20] that is used to choose the optimal inlier set. In Fig. 7, we use scatterplots to visualize the correlation between some of these features and pose recovery accuracy. Specifically, Fig. 7(a) plots the relationship between the confidence metric used to choose the optimal inlier set and location error of the pose estimation while Fig. 7(b) does the same for the number of inliers remaining after RANSAC homography and location error. The red (blue) dots in the scatterplots correspond to images with less (more) than 4 meters of location error. As seen, query images with larger location error tend to have less inliers and a smaller inlier set confidence metric.

We also perform support vector regression (SVR) on the training set and use the resulting regression model to predict location error of the testing set for our proposed pose recovery method. In doing so, we assign an arbitrarily large location error of 100 meters to the negative examples in the validation set. As seen in Fig. 8, there is a reasonable correlation between our predicted and actual location error.

We find the predicted binary label of the image retrieval and pose estimation confidence system to be in agreement with the actual groundtruth label 86% and 89% of the query images in the validation set respectively. Figs. 9(a) and (b) show the distribution of confidence values for image retrieval and pose estimation respectively on the validation set. Green (red) bars represent the confidence distribution of images whose predicted label (do not) match the groundtruth label. To create an overall system confidence score between 0 and 1 and prediction label, we use the following algorithm below:

```
overall confidence = 0.5 * (retrieval confidence + pose
confidence);
if overall confidence >0.5 then
    prediction label = true;
else
    prediction label = false;
end
```

By comparing the groundtruth and the overall confidence prediction labels for the query images in the validation set, the accuracy of the overall confidence estimation is determined to be 86%. Fig. 9(c) shows the distribution of overall confidence scores for the validation set.

To determine the optimal location error threshold for t_s, we experientially set it to values ranging from 1 to 12 meters and test the accuracy of the pose estimation confidence system. As shown in Fig. 10, the optimal value for the threshold is around 3-5 meters.

V. Experimental Results

For our experimental setup, we use the same database and query set that was used to evaluate our previous indoor localization pipeline [1]. We use the ambulatory human operated backpack of Fig. 2 to scan the interior of a two story shopping center located in Fremont, California. To generate the image database, we collect thousands of images with two 5 megapixel fish-eye cameras mounted on the backpack. These heavily

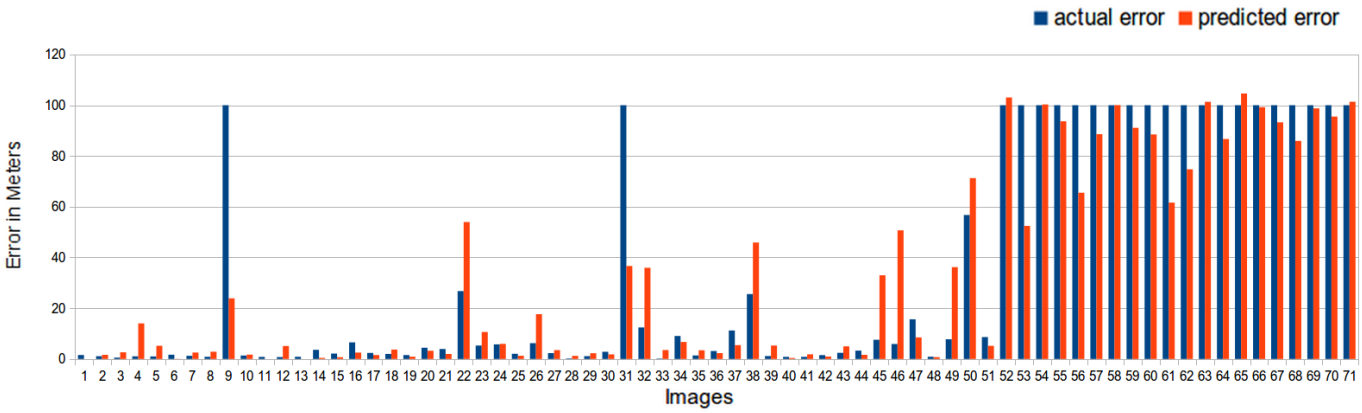


Fig. 8. Plot of actual (blue) vs predicted (red) location error for images in validation set using SVR regression. For the negative examples in the validation set, we set the actual error to be an arbitrary high value of 100 meters.

distorted fish-eye images are then rectified into 20,000 lower resolution rectilinear images. 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.

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 downsampling 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%. Detailed analysis of the failure cases reveal that two of the incorrectly matched query images correspond to a store that does not exist in the image database. Therefore, the effective failure rate of our image retrieval system is 3 out of 80 or less than 4%. As shown in Fig. 11(a), successful retrieval usually involves matching of store signs present in both the query and database images. In cases such as Fig. 11(b) where retrieval fails, there are few matched features on the query image's store sign. The image retrieval rate is unchanged from our previous pipeline in [1], since we have made no modifications to the retrieval process.

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 groundtruth the pose of each query image taken. Groundtruth is estimated by using the 3D model representation of the mall, and distance and yaw measurements recorded during the query dataset collection. We first locate store signs and other distinctive scenery of the query image within the 3D model to obtain a rough estimate of the query image pose, which is then refined using the measurements. The resulting groundtruth values are in the same coordinate frame as the output of the pose recovery step.

Fig. 12(a) compares the difference in depth values between sparse and dense depthmaps using a metric called absolute depth ratio (ADR). ADR refers to the ratio between sparse and dense depthmap values at the pixel location of a SIFT feature and ideally, should be close to 1. To ensure that the ratio can be averaged together for SIFT features in an image, we choose it to be always greater than 1; this means ADR is obtained by dividing the larger depth value by the smaller one. The mean ADR of an image is the average of all relevant SIFT feature ADRs. In Fig. 12(a), we plot the mean ADR of each matched database image. The red bars correspond to the mean ADR when considering all SIFT feature matches while the blue bars show the mean ADR corresponding to homography inliers. The mean ADR for all the SIFT feature matches of a database image is high, averaging around the 1.7-2 range, due to the presence of many SIFT features that are too distant from the camera and cannot be triangulated accurately. However, SIFT features on signs and posters in front of stores tend be to close to the camera and are usually accurately triangulated, causing most mean inlier ADRs to be close to 1. Since scale recovery only requires depth values at the pixel locations of homography inliers, both sparse and dense depthmaps produce similar translation vectors. An image by image comparison of localization accuracy and inlier mean ADR is seen in Figs. 12(b). For the majority of the images, the location error with either dense or sparse depthmaps are almost identical and not surprisingly, their inlier ADR values are close to 1. There are a few images namely 27, 32, and 72 for which the ADR is high, but the relative difference between sparse and dense location error is large as well. This behavior is expected since significant differences in inlier depth values should lead to significant changes in the scale of the translation vector.

Fig. 13 summarizes the performance of the pose estimation stage of our pipeline. Figs. 13(a) and (b) show the cumulative histogram of dense [1] and sparse depthmap location error respectively. As shown in Figs. 13(d), the location error of pose recovery for both dense and sparse depthmaps are quite similar, with only a slight decrease in accuracy for sparse depthmaps. Over 55% of all the images have a location error of less than 1 meter for both dense and sparse depthmaps. As expected, yaw error, shown in Figs. 13(c), is the same for both dense and sparse depthmaps. This is because depthmap

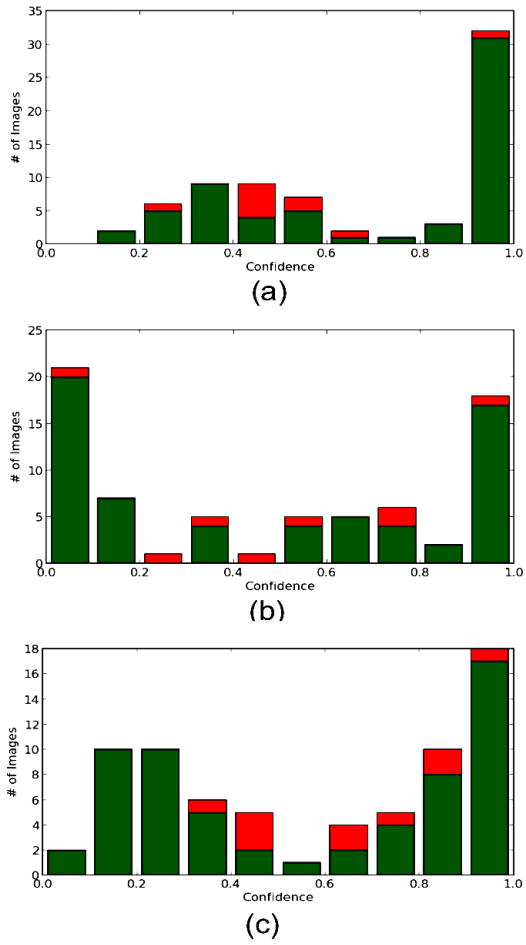


Fig. 9. Confidence distribution for (a) image retrieval, (b) pose recovery, (c) overall system on the validation set; Red (green) bars correspond to incorrectly (correctly) predicted images.

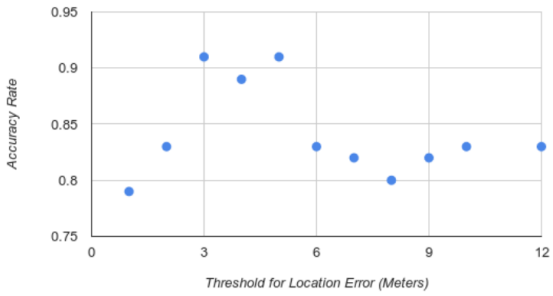


Fig. 10. Scatterplot showing relationship between pose recovery confidence estimation accuracy and the threshold t_s used for location error.

information is only used for scale recovery within homography while yaw is estimated separately during the vanishing point analysis stage of the pose estimation [20]. As seen in the example in Fig. 14(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 by the RANSAC homography [20]. Conversely, in less accurate cases of pose estimation where the location error exceeds 4 meters, the RANSAC homography finds "false matches"

Fig. 11. (a) Successful and (b) unsuccessful examples of image retrieval. Red lines show SIFT feature matches.

between unrelated elements of the query and database images. For instance in Fig. 14(b), different letters in the signs of the two images are matched. In general, we find that images with visually unique signs perform better during pose estimation than those lacking such features.

On a 2.3 GHz i5 laptop, our complete pipeline from image retrieval to pose recovery takes on average 10-12 seconds to run. On an Amazon EC2 extra-large computing instance, the runtime is reduced further to an average of 4.5 seconds per image. The individual runtimes for each image is highly variable, with some images taking twice as long as the average time.

VI. Conclusion

In this paper, we have presented a reduced complexity data acquisition system and processing pipeline for image-based localization in indoor environments with significant cost and time savings and little loss in accuracy. Several possible improvements to our image-based localization pipeline include tracking the position of the user and reducing the amount of noise in the depthmaps by utilizing more images for the sparse depthmap generation process. For future research, we planning to examine ways to further increase the accuracy of indoor localization. One method we are exploring is to combine our image-based indoor localization pipeline with a WiFi-based indoor localization system. The final position is determined by a particle filter that receives measurement updates from both localization systems.

References

[1] Jason Zhi Liang, Nicholas Corso, Eric Turner, and Avideh Zakhor, "Image Based Localization in Indoor Environments." in 4th International Conference on Computing for Geospatial Research and Application, 2013.

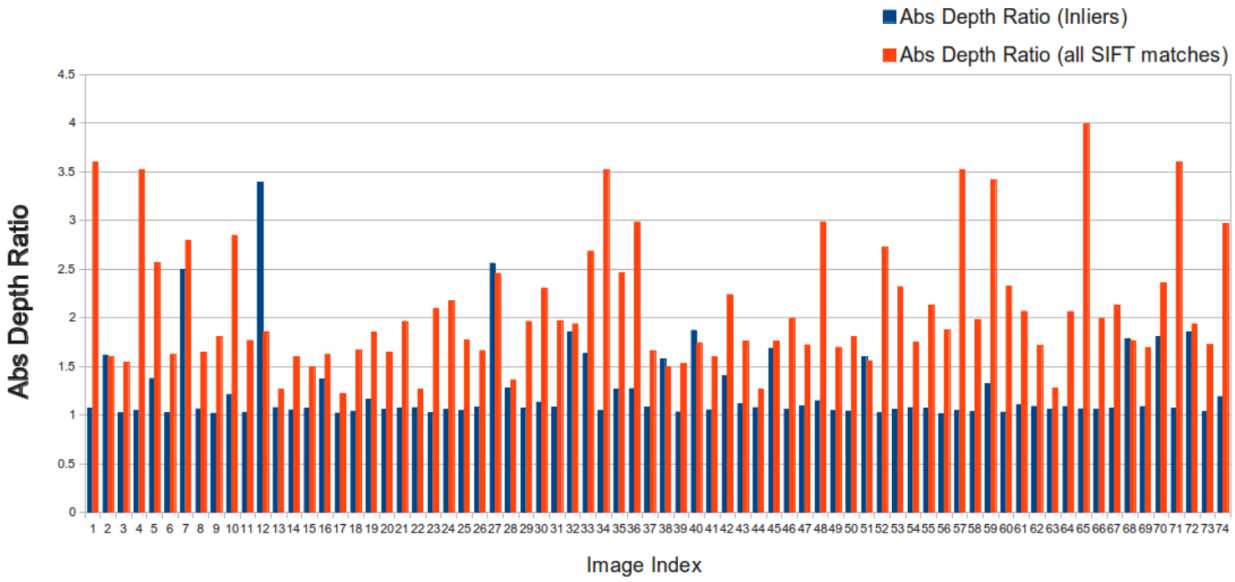
[2] L. I. U.Xiaohan, Hideo Makino, and M. A. S. E. Kenichi, "Improved Indoor Location Estimation using Fluorescent Light Communication System with a Nine-channel Receiver," in IEICE Transactions on Communications 93, no. 11, pp. 2936-2944, 2010.

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

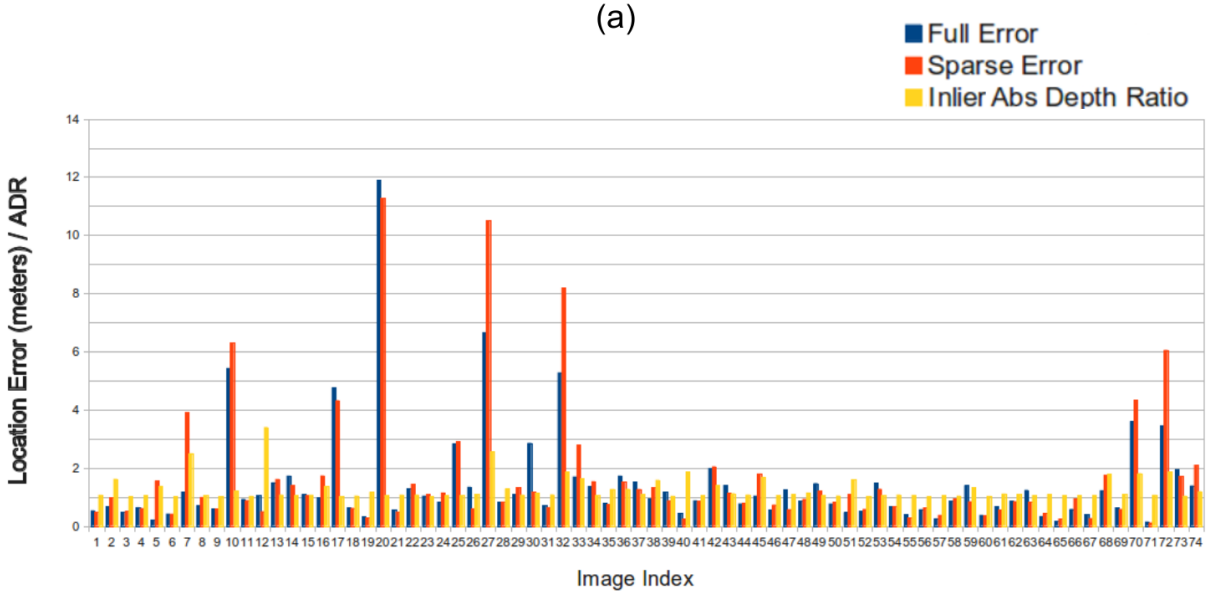
[4] Joydeep Biswas and Manuela Veloso, "WiFi Localization and Navigation for Autonomous Indoor Mobile Robots," in International Conference on Robotics and Automation, 2010.

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

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



(a)



(b)

Fig. 12. (a) Comparison of absolute depth ratio for all SIFT feature matches and homography inliers. (b) Comparison of absolute depth ratio for homography inliers and location error for both sparse and dense depthmaps.

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

[8] 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, 2011.

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

[10] 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.

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

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

[13] Richard Hartley and Andrew Zisserman, Multiple View Geometry, Cambridge University Press, 2004.

[14] 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.

[15] 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.

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

[17] Jerry Zhang, Aaron Hallquist, Eric Liang, and Avideh Zakhor,

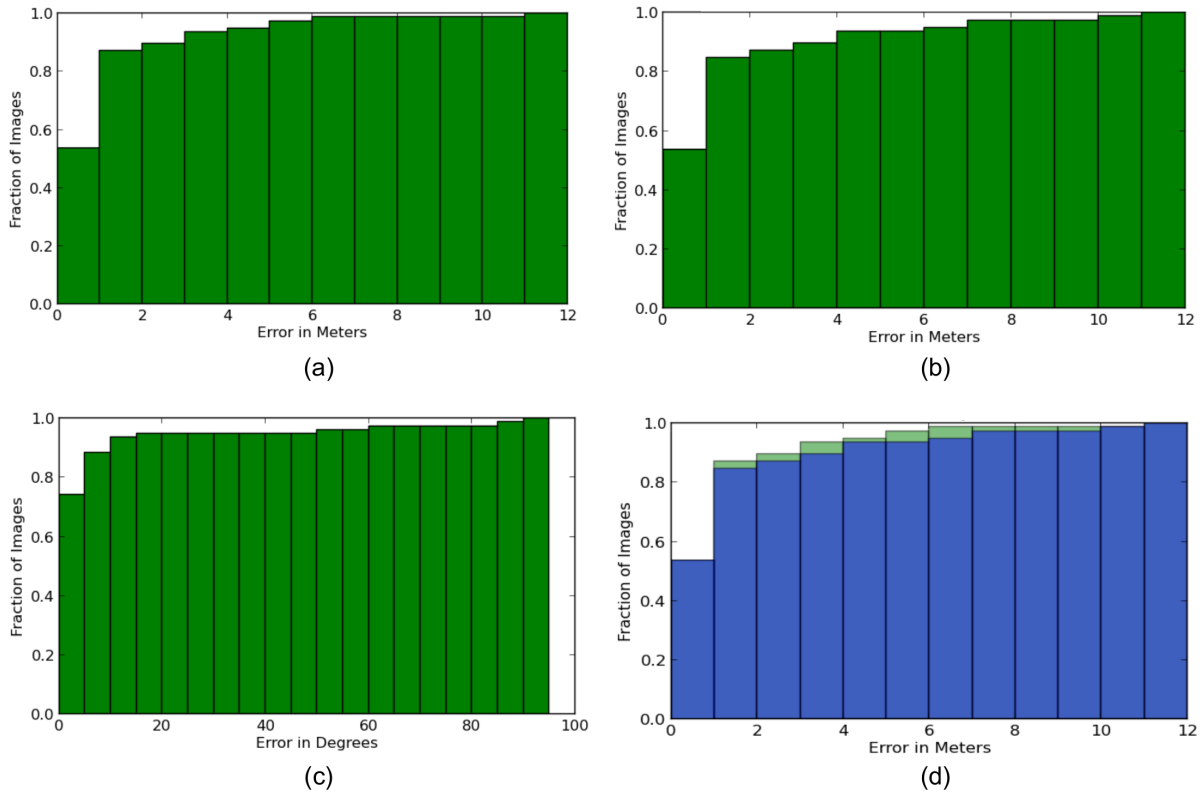


Fig. 13. Location error with (a) dense and (b) sparse depthmaps. (c) Yaw Error. (d) Comparison of dense and sparse location error. Dense depthmap results are shown with green and sparse depthmap results are shown with blue.

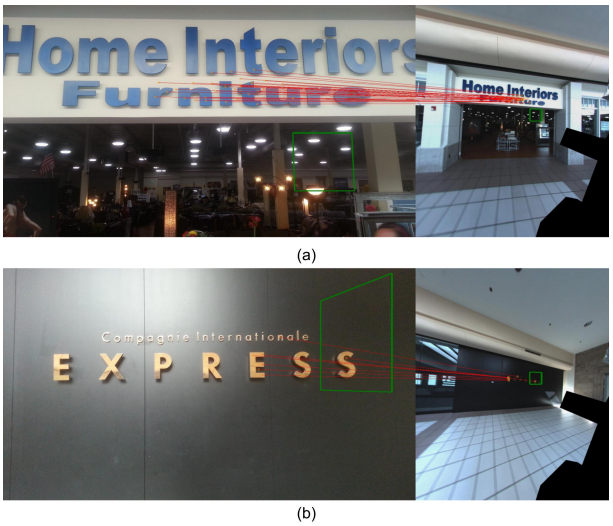


Fig. 14. (a) Example of accurate pose estimation on query image. (b) Example of inaccurate pose estimation. Notice how different letters in the same sign are matched.

"Location-Based Image Retrieval for Urban Environments," in International Conference on Image Processing, 2011.

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

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

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

[21] Eric Turner and Avideh Zakhor, "Watertight As-Built Architectural Floor Plans Generated from Laser Range Data," in Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, 2012.