End-to-end Image-based Indoor Localization for Facility Operation and Management

Yujie Wei[a] and Burcu Akinci[a]

[a]Department of Civil and Environmental Engineering, Carnegie Mellon University, United States of America
E-mail: yujiew@andrew.cmu.edu, bakinci@cmu.edu

Abstract -

Recent research on facility management has focused on leveraging location-based services(LBS) to assist on-demand access to model information and on-site documentation. Researchers highlight that fast and robust indoor localization is of great importance for location-based facility management services, especially considering when using facility management services in a mobile computing context. Despite the importance of location data, most existing facility management systems do not support LBS due to the following reasons: 1) Signal-based indoor localization methods, such as WIFI, RFID, Bluetooth and Ultrasound, require installation of extra infrastructure in a building to support localization, 2) Visual-based indoor localization methods, such as LiDAR and camera, still depend on feature point detection and matching that require heavy computation and can be impacted by environmental conditions. In this paper, the authors present an end-to-end image-based localization framework to support facility operation and management using a convolutional neural network (CNN). The proposed framework contains two modules: mapping and localization. The mapping module takes a set of training images with their pose as input and trains a CNN model for the localization module. The localization module takes a single image and the trained model as input and outputs estimated camera pose (position and view angle). Compared to conventional methods, the proposed end-to-end image-based indoor localization framework does not require any infrastructure installed in a building and can achieve real-time 6-DoF localization that is robust to different lighting conditions and scenes with poor texture. The proposed framework was evaluated on publicly available datasets and the results show that end-to-end imaged-based method can achieve real-time 6-DoF localization with acceptable accuracy and a small map size.

Keywords -

Facility Operation and Management; Image; Indoor Localization; Convolutional Neural Network; Deep Learning

1 Introduction

In this section, the authors give a brief introduction to location-based facility management and the status of current research.

1.1 Background

Locating and tracking people, equipment, and facility components within building supports many novel facility management applications such as asset monitoring [1, 2, 3], infrastructure inspection [4], cross-registration with BIM [5], and field reporting using mobile devices [6]. A location-based facility management system (LB-FMS) has a central geospatial database that stores facility information and allows all possible users to access or potentially change the data conveniently. LB-FMS is expected to streamline building operation activities, such as updating infrastructure status, documenting repair needs, managing work orders and inspection, through automatically linking the data collected by a mobile device at the scene to the facility management model. However, given the location of mobile device only, it's still challenging finding the correspondence at a component level, e.g, linking a window captured at the scene with its corresponding window element in the model. The primary concern that limits component-level registration is the lack of complete pose information (location and view angle). With location information only, there is an infinite number of possible ways to register a facility component to its model due to the freedom of rotation (Figure 1a). In comparison, with complete pose information, the correspondence could be easily established through ray tracing (Figure 1b). The resulting component-level correspondence can reduce the need for manually finding the corresponding component at the scene. In other words, an LB-FMS with complete pose information can be used for automating many of the aforementioned applications.

1.2 Previous Research

Current indoor localization approaches adopted in an LB-FMS can be generally divided into two categories: 1) Signal-based methods and 2) Visual-based methods. Signal-based methods, such as WIFI [7], RFID [8], Bluetooth [9] and Ultrasound [10], estimate locations by comparing recorded signal signatures with signals captured on site. For example, WIFI-based localization leverages the prior known router positions to estimate the signal

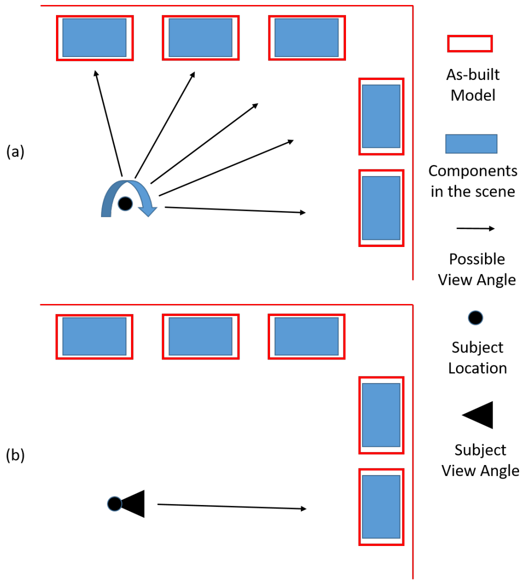

Figure 1. Finding corresponding as-built model for the object in a scene with locations only (a) vs. with poses (b)

strength at different locations within a building. When the client device receives signals from multiple routers, the position can then be estimated using triangulation. Other signal-based methods employ similar mechanisms as well. According to [11], signal-based methods have two primary limitations when applied in facility management applications: 1) They need to have the corresponding infrastructure, such as routers, beacons, RFID tags, or ultrasound receivers, installed in building to facilitate the localization service. 2) They can only output the location (X, Y, Z coordinates) but not the view angle (pitch, yaw, roll angles). Without the view angle information, locating facility components in the as-built model would be more difficult compared to the one with view angle information. Figure 1 demonstrates how pose information can reduce the search space for correspondence.

The visual-based localization methods, especially the image-based ones, are proposed to address the two limitations stated above. First of all, image-based methods do not need extra infrastructure to locate a person. Moreover, image-based localization methods can output location and view angle at the same time, making it much easier to estimate the location of what is being seen in a scene and linking it to what exists in a digital information model (such a 3D building information model). Typical image-based localization methods include image retrieval [12], direct 2D-3D feature matching [13], and end-to-end learning [14]. Image retrieval methods build a spatial database of a facility from geo-referenced images during mapping and retrieve the image that is most similar to the queried image for localization. Since the queried image is not likely to have identical poses (location and view angle) as the images in a database, image retrieval localization can only provide a rough estimation of the pose. Direct 2D-3D matching method reconstructs the 3D model using Structure-from-Motion (SfM) from images, and stores extracted feature points, such as Scale-Invariant Feature Transform (SIFT) in map (Figure 1). During localization, it compares the extracted feature points from the queried image to the ones in the database and estimates the camera pose using epipolar geometry. The localization accuracy can be very accurate (10-20 cm[10]) when feature point matching is successful. However, due to camera intrinsic differences, change of lighting condition and motion blur, direct 2D-3D matching could fail ungracefully even when a small subset of mismatched feature points exist. Another limitation of the 2D-3D matching method is computation complexity. An image usually has 300 to 500 feature points while a map is usually reconstructed from thousands of images. Therefore, locating one image requires finding the best match from millions of feature points which can take several seconds. Hence, existing 2D-3D matching method is also not scalable and might not be applicable for real-time localization needs.

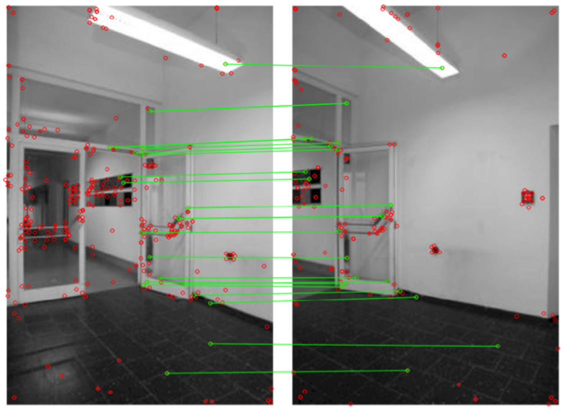

Figure 2. Feature-based Localization using SIFT Matching

Considering these limitations, the authors explored a new approach that learns to automatically extract useful features from images according to the desired objective (in our case, the objective is to minimize localization error), and that uses extracted features to estimate image poses. This approach is being referred to as end-to-end learning-based localization and Section 2 gives a brief introduction to it. Section 3 overviews how the proposed end-to-end image-based localization acts as an important module within a facility management system. Sections 4 and 5 describe the evaluation of the proposed approach with respect to its performance on a publicly available dataset and show the corresponding results obtained.

2 End-to-end Learning-based Localization

Convolutional Neural Network (CNN) [15] was originally proposed to address the image classification problem that can output a discrete label such as door and window given an image as input. CNN also works well in regression problems that require estimating continuous variables such as locations and view angles. In general, CNN is a powerful tool that can automatically learn how to extract useful features with respect to different objectives. End-to-end learning-based localization aims at training a CNN that maps an image to a pose. The localization process consists of two stages: mapping and localization. Basically, in the training (mapping) stage, the system takes a set of training images with ground truth pose and trains a Convolutional Neural Network(CNN) by minimizing the mean squared error(MSE) between the predicted pose and the ground truth pose. In the testing (localization) stage, the trained network takes an image as input and predicts the 6-DoF pose. The problem can be formally written as: given a set of images and their corresponding poses, find a function f that maps the input image $\mathbf{X}$ to its pose $\mathbf{y}$ through $\mathbf{y} = f(\mathbf{X})$ with the lowest localization error. The function f is defined by all the parameters of a CNN.

2.1 CNN Layers

A typical CNN architecture consists of several layers such as convolution, activation, transition, batch normalization (BN), and fully connection (FC). Below is a brief description of the functionality of each kind of layer.

2.1.1 Convolution

Convolution layers play a role of extracting features from input images in CNN. Given an image $\mathbf{X}$ (a tensor with size width × height × color channel), a convolution layer transforms the input by computing linear combination as below:

$$conv(\mathbf{X}) = \mathbf{w}^T\mathbf{x} + \mathbf{b} \tag{1}$$

where $\mathbf{w}$ is the weight vector of a convolution layer, $\mathbf{x}$ is the vector obtained from unrolling the input tensor $\mathbf{X}$, and $\mathbf{b}$ is the bias vector. With different weights, convolution layers are able to extract different features from the input that are useful for achieving the objective.

2.1.2 Activation

Activation layer provides non-linearity between two convolution layers so that the function model is not limited to linear space. A common choice of activation function is Rectified Linear Units(ReLU):

$$ReLU(\mathbf{x}) = max(\mathbf{x}, 0) \tag{2}$$

where $\mathbf{x}$ is the output from last layer.

2.1.3 Transition

Transition layer reduces the input size by averaging. For example, a 2×2 transition layer will average reduce an image with size $56 \times 56 \times 3$ to $28 \times 28 \times 3$. This is used for reducing the number of parameters the system needs to learn.

2.1.4 Batch Normalization

Batch normalization normalizes the input by subtracting its mean from the input and dividing it by the standard deviation. The goal of batch normalization is to improve the generalizability of the model.

2.2 CNN Architecture

In the proposed approach, the authors employ the DenseNet [16] structure for pose regression because of its strong capability of extracting visual features. Depending on the loss function, the extracted features can be used for both classification and regression task. In this problem, we changed the last FC layer to a Sigmoid layer for regression purpose. The adjusted FC layer contains 7 outputs, corresponding to 3 location coordinates(x,y,z) and 4 rotation coordinates(q_1, q_2, q_3, q_4 represented in quaternion format). Table 2.2 shows the DenseNet architecture. Notice that each conv layer listed in the table actually corresponds to a BN-ReLU activation-Conv sequence. The details of each layer can be found in [16].

2.3 Objective

One possible objective is to minimize the mean squared error (MSE) between the predicted pose and the ground truth pose as discussed in [14]. Specifically, denote the ground truth location as a translation vector $\mathbf{t} = [x, y, z]$ and the ground truth view angle as a rotation quaternion $\mathbf{q} = [q_1, q_2, q_3, q_4]$. Notice that the rotation quaternion can be converted back to the axis angle representation. Similarly, denote the predicted location and view angle as $\hat{\mathbf{t}}$ and $\hat{\mathbf{q}}$. The MSE loss of a single image $\mathcal{I}$ to be minimized can be represented as:

$$L(\mathcal{I}) = ||\mathbf{t} - \hat{\mathbf{t}}||_2 + \beta||\frac{\mathbf{q}}{||\mathbf{q}||} - \hat{\mathbf{q}}||_2 \tag{3}$$

where β is a hyperparameter that balances the rotation and translation error.

Table 1. DenseNet Architecture for Localization [16].

Layers	Output Size	DenseNet-121
Convolution	112×112	$7 \times 7 \times 2$ conv
Pooling	56×56	$3 \times 3 \times 2$ max
Dense Block(1)	56×56	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 6$ conv
Transition Layer(1)	56×56 28×28	1×1 conv $2 \times 2 \times 2$ avg
Dense Block(2)	28×28	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 12$ conv
Transition Layer(2)	28×28 14×14	1×1 conv $2 \times 2 \times 2$ avg
Dense Block(3)	14×14	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 24$ conv
Transition Layer(3)	14×14 7×7	1×1 conv $2 \times 2 \times 2$ avg
Dense Block(4)	7×7	$\begin{bmatrix} 1 \times 1 \\ 3 \times 3 \end{bmatrix} \times 16$ conv
Regression Layer	7×7 average pooling 7D fully-connected layer	

An alternative of the objective is to minimize the reprojection error [17]. Denote the ground truth 3D point as P, the reprojected 2D point as p, camera intrinsic (aperture, focal length, etc) as K, and the camera model as a transformation $\mathcal{T}$. The relationship between 3D points and 2D points can be represented as:

$$\begin{aligned} p &= \mathcal{T}_{K,\mathbf{t},\mathbf{q}}(P) \\ &= K[Q|\mathbf{t}]P \end{aligned} \tag{4}$$

where Q is the corresponding rotation matrix of $\mathbf{q}$ and N is the number of known 3D-2D matching pairs. The loss is then defined as:

$$L(\mathcal{I}) = \frac{1}{N} \sum_{P \in \mathcal{P}} \|\mathcal{T}_{K,\hat{\mathbf{t}},\hat{\mathbf{q}}}(P) - \mathcal{T}_{K,\mathbf{t},\mathbf{q}}(P)\|_2 \tag{5}$$

which aims at minimizing the reprojected error between the ground truth pose and the predicted pose. Though the second objective does not need the hyperparameter β, it requires a set of 3D points of the scene as prior, which is usually not available. Therefore, the authors employed the first objective in the proposed approach.

3 Overview of the Proposed Approach

To leverage the end-to-end learning-based localization in facility management, the authors propose an approach that contains four modules: mapping, localization, facility detection, and model update. The pipeline of the proposed approach is shown in Figure 3.

3.1 Mapping

The image-based mapping module takes images and their poses as the primary input but also accepts other

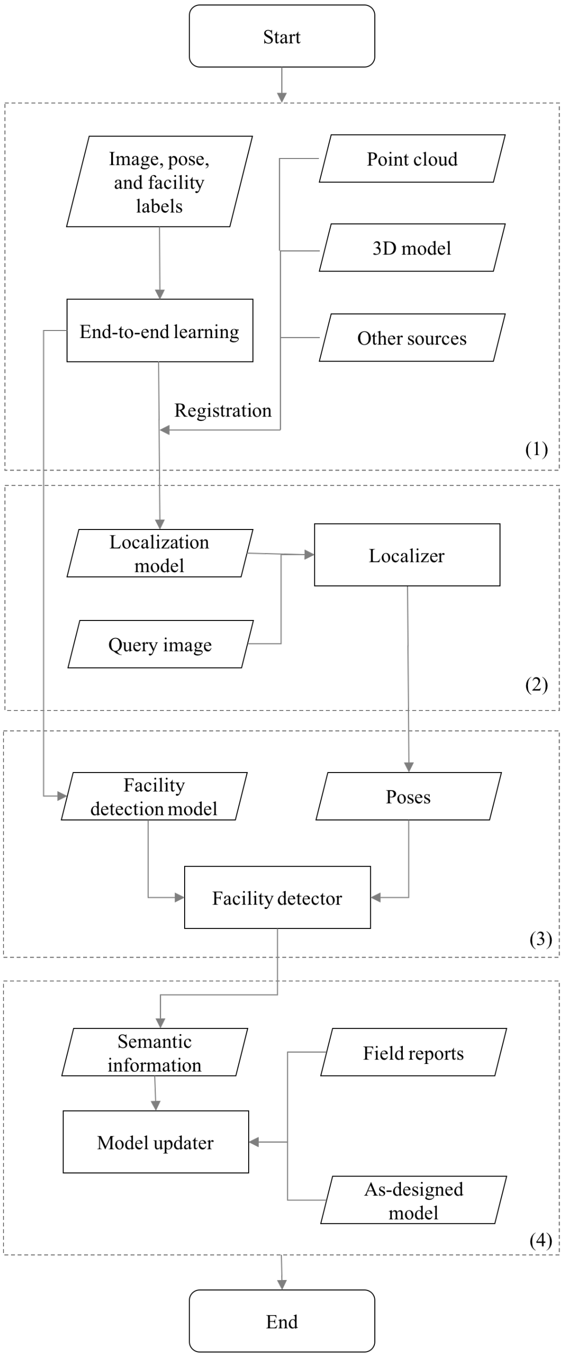

Figure 3. The Proposed Approach with Four Modules: (1) Mapping, (2) Localization, (3) Facility detection, (4) Model update.

.

data sources such as point cloud, 3D models, and a depth map. The output of the module is a trained network that can predict the camera pose given an image. As stated in Section 2, the goal of mapping is to find a function f with parameters $\mathbf{w}^*$ that minimizes the localization error on the training set:

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{M} \sum_{i=1}^{M} L(\mathcal{I}_i|\mathbf{w}) \tag{6}$$

where M is the number of images in the training set and $L(\mathcal{I}_i|\mathbf{w})$ is the error of the prediction on the i-th image $\mathcal{I}_i$ given parameters $\mathbf{w}$.

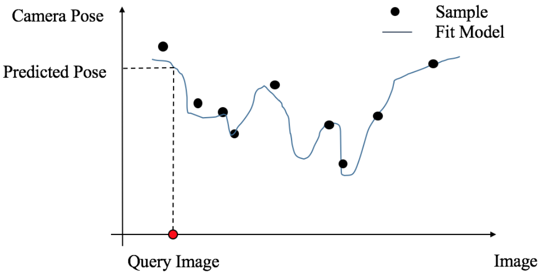

Figure 4. Simplified Representation of Mapping and Localization

.

Figure 4 shows a simplified representation of a mapping model where the x-axis is the image input, and the y-axis is the camera pose. Since the fitness function is non-linear, there is no closed-form solution of the optimization problem. Therefore, to find the optimal weights $\mathbf{w}^*$, we need to employ optimization methods such as gradient descend to gradually adjust the weights. The basic idea of optimization methods is to start from a random weight and moves along the direction where the loss function goes down. Usually, the data will be randomly split into three parts: a training set, a validation set, and a test set. The model is fit on the training set and usually stops optimization when the loss of validation set reaches the minimum. The test set is used for testing purpose only.

3.2 Localization

The image-based localization module (Part (2) in Figure 3) takes an image and the trained model from the mapping module as input, and outputs the 6-DoF localization result (x, y, z, pitch, yaw, roll) as shown in Figure 4. In the mapping(training) stage, the model is trained to extract features from images that are useful for determining image pose. To understand the mechanism of localization module, we can compare it with the conventional feature-based localization method. In the feature-based localization method, we manually extract the feature point from the image, match the feature point with the ones stored in a database, and estimate the location based on matching as shown in Figure 2. In contrast, end-to-end localization learns to extract relevant features by minimizing localization error. It does not require strict matching. Instead, the trained model tries to estimate the parameter of the function that maps an image to its pose.

Using end-to-end learning-based localization, the localizer does not need to detect feature points from the query image and conduct feature point matching. Instead, the pose is estimated purely by the image itself, which provides two advantages compared to the 2D-3D matching localization method. First, since the weights of the model and the query image have fixed size, the localization time is independent of the map size (Scalability). Second, CNN is able to output accurate pose even with downsized images. For example, to fit the training images into a single GPU with 11 GB of memory, a typical RGB image size used in CNN is $224 \times 224 \times 3$. In the contrast, the feature point detection requires high-resolution images as its input, such as RGB images with size $3000 \times 2000 \times 3$, to improve the feature point matching quality. The computation complexity of end-to-end learning-based localization is much smaller than the one of direct matching localization. Therefore, with proper choice of network architecture and image size, the localizer can provide real-time localization results.

With pose information, if the position of a facility component on an image is known, the component can be linked to a pre-registered as-built model through ray tracing as shown in Figure 5.

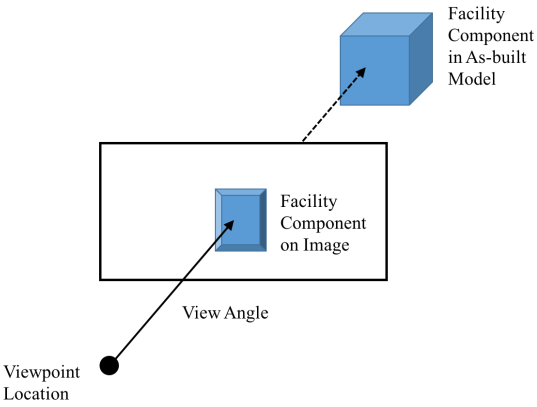
Facility Component in As-built Model
Facility Component on Image
View Angle
Viewpoint Location

Figure 5. Finding corresponding model for detected facility components on image using pose information

.

4 Implementation and Experiment

Section 4 discusses the implementation details and presents preliminary results on two publicly available datasets. The authors tested both the proposed method and the state-of-art Structure-from-Motion (SfM) methods using Colmap Library [18] on two datasets and compared their performance.

4.1 Dataset

The authors tested the end-to-end learning-based localization module on two different datasets. The first one is the Navvis dataset [19]. The Navvis testbed is located on the first floor of the main site of Technical University of Munich(TUM) with an indoor track length of 434m. The dataset consists of 3146 high-resolution DSLR images whose size is 3456 × 5184, point cloud model of the scene, as well as the ground truth camera pose of each image. The ground truth provides the transformation matrix that can be decomposed into locations and view angles. The dataset also provides the camera intrinsic for 3D reprojection. The second dataset is the Baidu IDL indoor localization dataset [20]. The IDL dataset is captured at a mall, containing 682 training images captured from DSLR camera and more than 2000 query images captured from other cameras. In the experiment, the authors randomly split the dataset into three parts: 8/10 for training, 1/10 for validation, and 1/10 for testing. The training set is used for optimizing the parameters of the model, while the validation set is used for determining when to stop training as well as choosing the proper hyperparameter β. Then the authors evaluate the proposed approach on the test set.

4.2 Implementation

4.2.1 Preprocessing

Before training the model, the authors conducted the following preprocessing over the data:

1. Downsize all images to 256 × 384 × 3 to accelerate training and localization. Compared to raw images, our GPU can process 64 compressed images in parallel.

2. Crop the image into four corners and a center using the size 224 × 224 × 3. The authors did the cropping for two reasons. First, it is an effective way of data augmentation that allows the network to have more data for training. Second, the authors used the weights of a pretrained network on ImageNet to initialize the model, images on ImageNet uses the same size as well. Notice that cropping will not change the ground truth pose.

3. Randomly add 10% lighting noise (AlexNet-style PCA lighting noise [21]) to make the model more robust to different lighting conditions.

4. Normalize the image using the mean and standard deviation of ImageNet. Normalization guarantees that the same range of values for each of the inputs to the model, which can effectively prevent ill-conditioned model and accelerate optimization process.

4.2.2 Training

The authors implemented the proposed model using PyTorch. The pretained densenet-121 model from ImageNet is employed as the base net in the experiment. The last layer of DenseNet was replaced by a fully-connected layers that has 7 outputs where the first 3 outputs are x, y, and z, while the last 4 outputs are q_1, q_2, q_3, q_4. As mentioned in section 2.2, the objective function has a hyperparameter β that balances the location and the view angle error. In the experiment, we follow the suggestion in [14] and set the $\beta = 150$ for indoor scenes. The optimization method is Adaptive Moment Estimation(Adam) [22] and set the learning rate as $1e-4$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ using the recommended settings utilized in [22]. The batch size is 64 which uses up to 11 GB of GPU memory. The model was trained on a desktop with Intel i7-7700k CPU, 32G RAM, and a Nvidia 1080Ti GPU.

4.3 Results

The performance of the localization module was evaluated from three perspectives: 1) accuracy, 2) robustness, and 3) efficiency. The average location error is $2.04m$ and the average view angle error is $11.3°$ on Navvis dataset, and $1.02m$ and $4.2°$ on IDL dataset. Figure **??** shows the localization result on Navvis dataset. In comparison, the SfM method failed to reconstruct a 3D model for Navvis dataset due to poor textures of the scene. On IDL dataset, the mean displacement and orientation error of the SfM method are $7.21m$ and $18°$ respectively due to the reconstruction ambiguity. Figure 7 shows the localization result on IDL dataset using SfM (a) and the proposed method (b). As shown in Figure 7, the SfM localization result has an ambiguity issue which is caused by incorrect feature point matching.

Figure 8 shows the translation and orientation error distribution respectively. On the IDL dataset, about 88% of the images has a translation error lower than $2m$. On Navvis dataset, about 93.3% images have a location error lower than $4m$. In comparison, the SfM method failed to reconstruct a 3D model on Navvis dataset and presented an ambiguity issue on IDL dataset. The reconstruction failure was due to the poor texture of the scene and low overlapping between images. In our experiment, the image

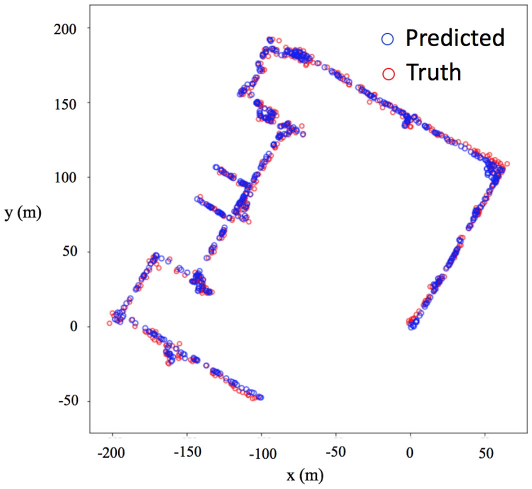

Figure 6. Localization Result using the proposed method on Navvis dataset.

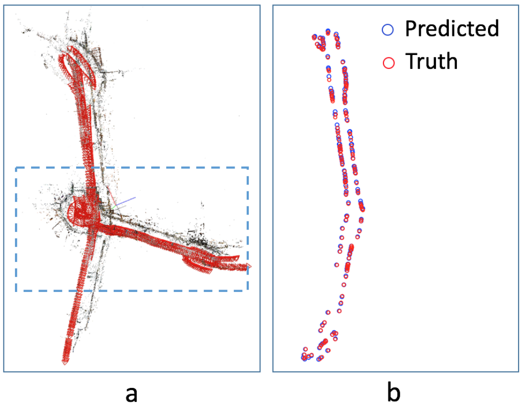

Figure 7. Localization Result from SfM (a) and the proposed method (b) on IDL dataset. The blue rectangle in (a) shows the reconstruction ambiguity.

registration rate on Navvis dataset is about 15%, which is not sufficient for 3D reconstruction. The ambiguity issue comes from the repetitive scenes in the indoor environment. When a scene presents similar pattern at different locations, incorrect feature matching will lead to ambiguity issues. According to the experiment results, the proposed method is robust to the aforementioned problems.

Regarding time and storage efficiency, during localization, predicting each query image takes averaged 0.04s on the aforementioned desktop using the proposed method. The map size of the proposed method is just the network itself, which is 35MB in total. In comparison, the state-of-art feature-based image localization method implemented in [23] needs an averaged 0.3s to finish localization in a map with only 900 registered images. Also, with 900 registered images and 400 feature points on each image (Each feature point is represented by 128 floating point numbers), the map size of the feature-based method reaches 175 MB and it will increase as the number of registered image increases. The fixed map size of the proposed method makes it more suitable for providing location services in a mobile computing context considering the limited memory resources of mobile devices.

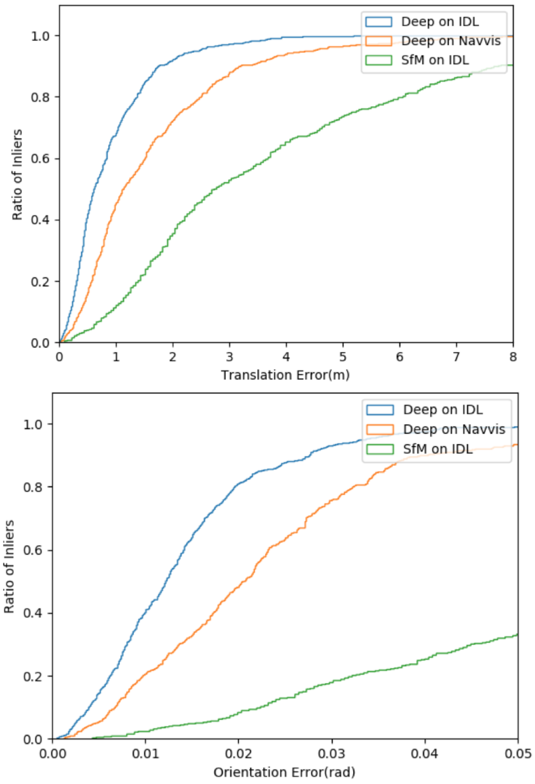

Figure 8. Translation and orientation error distribution on the testset of IDL dataset

5 Conclusion

In this paper, the authors explored an end-to-end image-based indoor localization method and proposed an approach of integrating it into a general facility management framework. Compared to signal-based localization methods, image-based methods do not require special infrastructure installed in the building and can output 6-DoF poses. Compared to conventional image retrieval and feature-based localization methods, end-to-end localization is robust to poor texture and repetitive scenes. Though feature-based methods can achieve high accuracy given accurate feature matching results, they are susceptible to inaccurate registration and reconstruction ambiguity as shown in the experiment. Moreover, end-to-end localization has a fixed map size and can provide real-time locations while the map size of feature-based methods will grow over time. Therefore, compared to feature-based methods, the proposed approach is more suitable for running on mobile devices such as cell phones or tablets.

In conclusion, end-to-end image-based localization is an alternative of indoor localization methods. It has the potential of being integrated with a facility component detection module to support facility management. However, there are also a few challenges when using end-to-end image-based localization. First, it requires ground truth poses (usually captured from a laser scanner) as input for training, which can be hard to capture in the industry due to the difficulty of cross-sensor calibration. Second, the performance of the proposed approach might be affected by image quality, over-lapping between images, texture richness, lighting conditions, and camera intrinsic difference. The authors will continue to evaluate the end-to-end image-based localization considering these variances and integrate it with the facility component detection module in future research.

6 Acknowledgements

The project is funded by a grant from the National Science Foundation (NSF), 1534114. NSF's support is gratefully acknowledged. Any opinions, findings, conclusions or recommendations presented in this paper are those of authors and do not necessarily reflect the views of the NSF.

References

[1] Ali Motamedi, Mohammad Mostafa Soltani, and Amin Hammad. Localization of rfid-equipped assets during the operation phase of facilities. Advanced Engineering Informatics, 27(4):566–579, 2013.

[2] Jochen Teizer and Patricio A Vela. Personnel tracking on construction sites using video cameras. Advanced Engineering Informatics, 23(4):452–462, 2009.

[3] Ming Lu, Wu Chen, Xuesong Shen, Hoi-Ching Lam, and Jianye Liu. Positioning and tracking construction vehicles in highly dense urban areas and building construction sites. Automation in construction, 16(5):647–656, 2007.

[4] Matthew M Torok, Mani Golparvar-Fard, and Kevin B Kochersberger. Image-based automated 3d crack detection for post-disaster building assessment. Journal of Computing in Civil Engineering, 28(5):A4014004, 2013.

[5] Khashayar Asadi Boroujeni and Kevin Han. Perspective-based image-to-bim alignment for automated visual data collection and construction performance monitoring. In Computing in Civil Engineering 2017, pages 171–178.

[6] Hyojoon Bae, Mani Golparvar-Fard, and Jules White. Image-based localization and content authoring in structure-from-motion point cloud models for real-time field reporting applications. Journal of Computing in Civil Engineering, 29(4):B4014008, 2014.

[7] Chin-Heng Lim, Yahong Wan, Boon-Poh Ng, and Chong-Meng Samson See. A real-time indoor wifi localization system utilizing smart antennas. IEEE Transactions on Consumer Electronics, 53(2), 2007.

[8] Guang-yao Jin, Xiao-yi Lu, and Myong-Soon Park. An indoor localization mechanism using active rfid tag. In Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006. IEEE International Conference on, volume 1, pages 4–pp. IEEE, 2006.

[9] Anja Bekkelien, Michel Deriaz, and Stéphane Marchand-Maillet. Bluetooth indoor positioning. Master's thesis, University of Geneva, 2012.

[10] Patrick Lazik, Niranjini Rajagopal, Oliver Shih, Bruno Sinopoli, and Anthony Rowe. Alps: A bluetooth and ultrasound platform for mapping and localization. In Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, pages 73–84. ACM, 2015.

[11] Saurabh Taneja, Asli Akcamete, Burcu Akinci, James Garrett, Lucio Soibelman, and E William East. Analysis of three indoor localization technologies to support facility management field activities. In Proceedings of the International Conference on Computing in Civil and Building Engineering, Nottingham, UK, 2010.

[12] Jürgen Wolf, Wolfram Burgard, and Hans Burkhardt. Robust vision-based localization by combining an image-retrieval system with monte carlo localization. IEEE Transactions on Robotics, 21(2):208–216, 2005.

[13] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 667–674. IEEE, 2011.

[14] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In Proceedings

of the IEEE international conference on computer vision, pages 2938–2946, 2015.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[16] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[17] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[18] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[19] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, and E. Steinbach. TUMindoor: An extensive image and point cloud dataset for visual indoor localization and mapping. In Proc. of the International Conference on Image Processing, Orlando, FL, USA, September 2012. Dataset available at http://navvis.de/dataset.

[20] Xun Sun, Yuanfan Xie, Pei Luo, Liang Wang, and Baidu Autonomous Driving Business Unit. A dataset for benchmarking image-based localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7436–7444, 2017.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc., 2012.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[23] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(9):1744–1756, 2017.