UNIVERSITY OF VIRGINIA
SCHOOL OF ENGINEERING AND APPLIED SCIENCE

INDOOR PLACE RECOGNITION FOR SITUATION AWARENESS USING DEEP LEARNING MODELS AND 3D POINT CLOUD COMPUTATION

A dissertation submitted in partial satisfaction of the
requirements for the degree of

Master of Science

in

SYSTEMS ENGINEERING

by

Amir Ashrafi

August 2021

The Committee Members:

Professor Devin K. Harris, Chair

Professor Arsalan Heydarian, Advisor

Professor Tariq Iqbal

Dean Jennifer L. West

Table of Contents

List of Figures

List of Tables

Abstract

Indoor Place Recognition for Situation Awareness Using Deep Learning Models and 3D Point Cloud Computation

by

Amir Ashrafi

With the increasing number of sensing devices in smart buildings for temperature, humidity, air quality, etc. acquiring information with regards to the state of the building and the location where a user is present has become very important recently in multiple domains including design, construction, facility management, and emergency response. This concept has been defined as Situation Awareness (SA) which can be obtained and improved by visualizing the information properly. The very first step to visualize such information is to identify the accurate location and orientation of the user in the indoor space. For accurate indoor localization, we are proposing to use a 3D point cloud model to train an end-to-end deep neural network which then is implemented on a head-mounted Augmented Reality (AR) platform (Microsoft HoloLens) to spatially localize users within an indoor space. To achieve this, a point cloud 3D model with more than 190 million points of a 17,000 sq. ft. open office space with diverse specifications such as office rooms, hallways, open areas, and crowded spaces were collected and registered by laser scanners. After pre-processing and generating submaps with different sizes, they

are ultimately turned into global descriptor vectors to match with the input queries that come from the user's ambient scanning with the HoloLens. Also, a real-time framework is designed to run on the HoloLens in which it captures the scanned 3D mesh around the user in real-time then sends the query to the neural network model by utilizing RESTful web service. Finally, the best sub-map candidates that match the queried 3D point cloud will be fetched to localize the user accurately. As result, we were able to reach 90% recall for the average top 1% of the results in the range of 1-meter threshold.

Acknowledgments

I would like to thank my advisor Professor Arsalan Heydarian very much for the support and understanding over these past three years. This thesis would have never been accomplished without him believing in me.

Chapter 1

Introduction

Decades have passed since the term Situation Awareness (SA) has been introduced from aviation psychology for addressing the operational functions of pilots during the flight in terms of their understanding from a dynamic environment [12]. SA could be translated into the cognitive activities that are the necessary inputs for decision making and increase the overall performance [12, 13] of individuals. [33] defines situation awareness as:

"Continuous extraction of environmental information, integration of this information with previous knowledge to form a coherent mental picture, and the use of that picture in directing further perception and anticipating future events."

In other words, first we need to know where we are located, compare the real-time spatial information with historical information, and finally plan the future steps.

Endsley [13] introduced the general SA flowchart which describes the set of information needed to create SA and how factors and operators connect to the SA cycle. We adapted the chart and slightly changed it based on the context and objectives of this research which is shown in figure1.1. In this chart, the surrounding environment perception is shown as the first step. After data acquisition, useful knowledge will be put together to show the user what is happening around them. Then, based on the information requested, future projection of environment status could be taken into account in order to help the operator effectively make decisions within the detected situation. Finally, the performance could be measured and proper feedback will be inserted into the environmental conditions. For instance, consider a user walks into an unknown building, for example an airport. The first step is the perception of the surrounding environment that translates the spatial features into localization information as well as identifying any special contextual features. In this example, a user may use the mobile device to capture the spatial features in order to identify his/her location in the terminal and receive relative information about their surrounding gates and airport facilities. In the meanwhile, SA system measures the performance of the application and the user to adapt itself to the environment.

In this research, we study the initial steps into SA for indoor environments; specifically, we introduce a robust SA framework for individuals in an indoor envi-

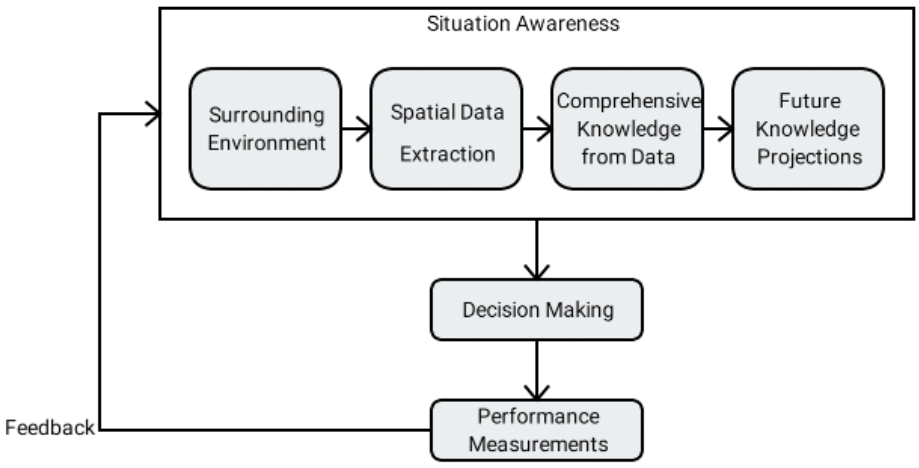


Figure 1.1: Situation awareness framework adapted and slightly changed from [13] based on our designed system.

ronment who intend to acquire spatial information as they walk through an indoor space. For instance, a user enters a new building and is trying to learn more about the design of the space, the residents of that building, and identify different areas where he/she can meet certain people. To achieve this, there is a need for a robust situational awareness system to identify where the user is located within the space and provide virtual information according to his/her spatial coordinates. This system could be generalized into various applications that involve data extraction associated with the indoor spaces of buildings. For example, first responders could benefit from having access to different building systems and sensor information to quickly localize occupants in need or wayfinding applications in new

buildings that they have never been inside before [8]. Similarly, facility managers could benefit from such SA systems in identifying building system components, access real-time and historic sensor information for different systems (e.g., HVAC, lighting), or visualize previous work-orders or design documents [18, 6, 38].

The first state of the SA cycle as shown in the figure 1.1is to perceive the surrounding environment. In order to accomplish this, we need to accurately identify the location of an individual within the indoor space. The most common tool being used for localization is GPS in outdoor environments. However, outdoor localization techniques that vastly rely on the Global Positioning System (GPS) cannot be used within an enclosed space with a ceiling because the satellite signals will be sabotaged by the obstacles. There is a vast number of applications of indoor localization which include reliable and effective methods to localize occupants, devices, or mobile robots in a building. Therefore, other methods have been proposed to improve indoor localization. Signal-based methods have significantly improved over the past years to identify the user's approximate location; however, there exist some limitations with these methods such as errors that are accumulated over time as a result of the presence of obstacles (e.g., interior walls) or limited access points in large indoor environments. In addition, it's well implied that installing the hardware equipment (access points such as routers, Bluetooth beacons) throughout the building is expensive and requires constant maintenance

[41]. To address these limitations, vision-based approaches have shown promising results[25]; however, these methods are not robust enough in some situations in which there are too many occlusions in the scene. Furthermore, the majority of the presented vision-based approaches only consider 2D images without considering the depth information which helps to detect occlusions and neutralize the effects of lights. Also, with emerging 3D point cloud building models that are produced during the design and construction of buildings with different tools such as drones and stationary/hand-held laser scanners, we are provided with the opportunity to build localization models that do not require excruciating on-site 2D image capturing to make the database in order to be used in deep learning. Besides, a 3D point cloud is more precise than a 2D image in terms of having depth information to avoid occlusions, invariance against lighting conditions, and picture quality.

In this work, we are proposing to use a 3D point cloud model with more than 190 million points of a 17,000 sq. ft. open office space to train an end-to-end deep neural network model which then is implemented on a head-mounted Augmented Reality (AR) platform (Microsoft HoloLens). We chose to utilize a head-worn AR device because of valuable opportunities it gives us such as, visualization benefits to improve productivity and comfort, live feedback both visually and stereo audibly to help to make decisions fast and effective, and futuristic modality of this

device which could be a prototype phase for advanced technology in the future. To achieve this, we trained our dataset which includes more than 40000 sub-maps extracted from the 3D point cloud which are ultimately turned into global descriptor vectors to match with the input queries from the user. A real-time system is then designed to be running on the HoloLens which is based on the user's location where spatial data could be fetched and shown to the user. To accomplish this, we have developed a query-based system implemented on HoloLens, which includes four cameras capable of capturing real-time 2D and depth information from an indoor space. This system captures the scanned 3D mesh around the user in real-time and matches it to the best sub-map candidate from the 3D point cloud. Then the result would be the user's position coordinates relative to the 3D point cloud coordinate system.

Our objective in this work is to utilize 3D point clouds for localizing the user in an indoor environment. The selected space to evaluate the proposed approach includes different levels of complexity such as different architectural and structural design features, different types, colors, and shapes of furniture and objects, as well as different size and use cases of spaces which could be challenging. We have addressed these issues by (1) fine-tuning the 3d point cloud place recognition models, followed by testing and validating the novel neural network models on a

3D point cloud generated from an AR headset (HoloLens 2). (2) Then through a case study, we present how we can refine the developed models for applying to the Microsoft HoloLens 2. This system could be used in numerous applications such as facility management, construction, tourism, or EMS (Emergency Medical Services) to successfully localize the user within indoor environments which have various complexity in terms of interior designs, occupancy levels, occupant activities (movement), etc. After localization, accessing building information locally such as sensors data and navigation services through augmented reality devices could help the user get the perception of their environment through visualization more thoroughly and comfortably especially in Microsoft HoloLens. The figure 1.2shows a prototype of our visual assistance application that runs on the HoloLens consisting of visualization of the sensors data that are being extracted based on the User's location as well as the visual location on the mini-map.

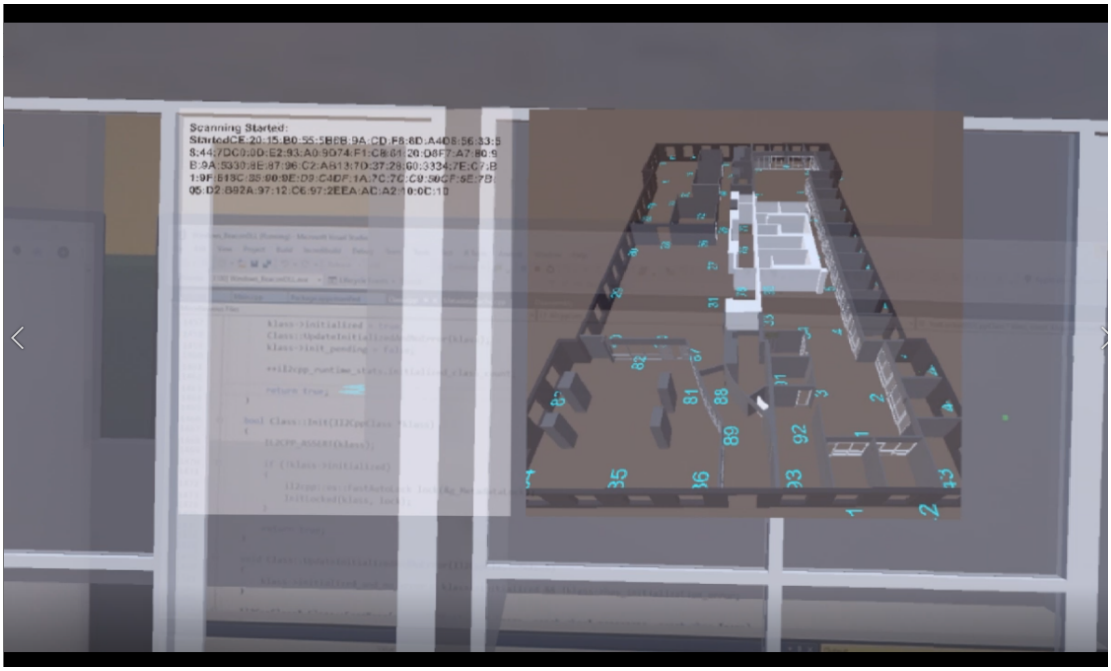


Figure 1.2: A prototype view of our Microsoft HoloLens application that shows the sensor's information in the left panel and the location of those sensors in the right panel after localization.

Chapter 2

Related Works

In this section, at first, we discuss the indoor localization literature such as visual recognition methods such as 3D BIM (Building Information Model) or point cloud models. We then review how augmented reality technologies are studied to facilitate localization and visualization in indoor applications. These applications exist in various fields, for instance, in design and construction and facility management, [3] assessed some works that have been done for supporting the operator on-site to connect the as-is conditions to previously collected 3D model. This would help to update the missing data depicted in the model by updating the information periodically and provide associated spatial data related to the place that a user is located for example in maintenance and repair operations.

For accomplishing this connection between the current status and known knowl-

edge, we need to start with indoor localization techniques that have been proposed over the past decades since GNSS (Global Navigation Satellite System) is not reliable and accurate for geo-referencing in an indoor environment [40]. [31] categorized three main technologies for indoor localization which consists of signal-based (WiFi, ultra-wideband, Bluetooth, RFID, ultrasound), motion-based (IMU), and image-based. Signal-based methods rely on installing a hardware infrastructure to make localization possible because of required access points to fulfill coverage of the environment[41]. Regarding the methods of signal-based localization which were implemented with the help of an augmented reality platform, [2] utilized Wi-Fi RSSI (Received signal strength indication) to train RSS fingerprint data on an RNN (P-MIMO LSTM) model. In this work, AR was used to align the floorplan on the real scene for creating a grid to collect the RSS data based on its coordination. Also, using signal strength data coming from RFID devices [36] or Bluetooth beacons [35] to estimate the user's location on a room-level scale; however, these approaches have shown to not be accurate and reliable because of signal fading resulting from passing through walls and obstacles which cause harsh signal strength fluctuations [41]. Also, because of lacking a user's visual angle view, the signal-based methods are not suitable for estimating the user's orientation. The motion-based method which is mostly based on IMU sensors (gyroscope, accelerometer, and magnetometer), is another approach to track the

wearable relatively unlike the other localization methods that are absolute [16]. Therefore, this method is usually fused with other approaches such as Wi-Fi or image-based methods to increase the accuracy of identifying the user's location and orientation [40].

Image-based or visual place recognition methods are the most commonly studied area of research. [25] classified these methods based on environment data (marker/camera pose, image/feature database, and 3D model), sensing devices (static and mobile cameras plus other sensors), detected elements (artificial markers, real features), and localization method (traditional image analysis, artificial intelligence).

The first approach is using natural/artificial markers (ArUco markers, indoor signs) to localize the user based on pre-known marker coordinates located in different areas in the indoor space. [17] presented a marker-based method for indoor localization to overlay a virtual room-scale model on the real room scene through an augmented reality device (HoloLens) in order to visualize in-situ information. In another work, [10] introduced analyzing the video frames with well-known markers such as room names or signs. They used a hand-held AR device to capture and visualize the augmented information on the screen. However, it's not clear how the user's relative position to the marker and the room was determined in the paper. Furthermore, this method cannot be generalized for all indoor envi-

ronments, especially those with different natural or artificial signs (e.g. Exit signs or different ArUco markers) [25].

The second approach is marker-less which divides into two main branches for place retrieval representations; hand-crafted representations (traditional) and deep-learning representations [24]. In hand-crafted representation, we have two main categories; local descriptors and global descriptors representation [24]. Local descriptor focuses on the patterns that happen in a path of the image and tries to highlight ones that differ from the neighborhood. On the other hand, a global descriptor tries to encode the integrated characteristics of an image as a whole[24]. Within the deep-learning representations, [37] proposed weighted parallel ICP (iterative Closest Point) to first speed up the ICP for dataset and model processing, second dividing the point cloud into two groups, line, and corner. Their method includes detecting features from corners and lines then matching those with the 2D blueprint's viewpoint. Nonetheless, it would be problematic in complex indoor spaces that do not have defined corners or edges for example rounded designs or with discontinuous patterns. [1] utilized deep CNN (Convolutional Neural Network) for training synthetic images obtained from the indoor 3D models to regress the camera pose and location. The device for this approach is a hand-held camera that sends query images to the model to find the camera pose and orientation. Their deep network model is trained over virtual trajectories that went through

the BIM and created synthetic images for the learning process. Although effective, the authors reported a number of shortcomings such as image quality could impact the result including blurriness or taking images from turning points. In addition, due to the changes in the environment over time such as changes in furniture types or their location, 3D BIM was shown to not be robust enough to be updated easily and effectively in terms of adding or removing the real changes in a building.

In another study [38] introduced a DenseNet CNN approach for segmentation and localization at the same time by entering one image as a query then associating a digital twin which consists of a digital building model that has the building's components to store the data. The segmentation part helps to detect the facility to connect it to the digital component and the camera pose is used for localization. However, This approach would accumulate the noise within the data generated from the segmentation and localization approaches and impacts on the results. For example in bad lighting conditions, segmentation error could be increased. Regarding local data association, [6] made an AR use case for facility management in which the system sent an image query to a CNN-based model, to retrieve the user's perspective and orientation from a 3D BIM model. As result, the matching image would connect to a 2D blueprint for acquiring the pipes to overlay on the floor. The accuracy of the system is reported as check-marked detected regions in

which the pipes visualizations are assigned. This approach could be challenging where two places are similar in structure or if the BIM model is not fully accurate to represent the structural and architectural features as well as the dynamic objects (e.g. furniture, crowd) in the real images. In Another study, [39] proposed a multi-modal approach by combining the Wi-Fi and image-based methods to locate the user. In the training phase by using indoor geometric reasoning, a 2D floorplan was extracted from an input image as well as a point cloud using SfM (Structure from Motion). Then by matching the point cloud and the floorplan, a database of the POIs (Place of Interest) was created. Finally, in the operating phase, first rough location estimation occurs with a Wi-Fi signal to reduce the search location then by getting an image from the scene, the candidate PCs are fetched from the database. However, the user should be aware of how they are capturing the image in making the dataset (there should be a flat facade in the view). Besides a Wi-Fi infrastructure should be available for the start, and similar to previous Wi-Fi methods, random fluctuation of the signals due to occlusions and noises would affect the rest of the system.

To address the challenges mentioned above, we are proposing to use point cloud as the reference and query input to our model in order to localize the user with respect to the global coordination of the building. This would alleviate the problems that 2D images are susceptible against such as, light impact, the image

Figure 2.1: Microsoft HoloLens 2

quality, user's condition, and depth information that are in the way of connecting the real user's viewpoint to the building 3D model. On the other hand, in 3D scanning, we are able to build this information by using Microsoft HoloLens spatial mapping capability in which utilizes the spatial anchors to remember the same place in real-time. This feature not only helps to reduce the amount of effort in scanning for a query, but also helps to update the changes in the scene.

Chapter 3

Methodology

Framework

We utilized the Microsoft HoloLens 2 to implement this indoor place recognition model for testing the real data collected from a user walking around an indoor space. Microsoft HoloLens 2 includes multiple sensors such as IMU, ToF depth, Infrared camera, visible light camera, and regular camera for recording. However, the most important aspect of this AR platform is its ability to do the spatial mapping from the surrounding area. This feature allows the device to create a 3D mesh from the space and build a world anchor reference for future access. In the presented case study, a user would wear the device and walk through an indoor open office space. By scanning the area with a spatial mapping feature, we could gather real-time meshes to build an AR-based 3d building model as our testbed

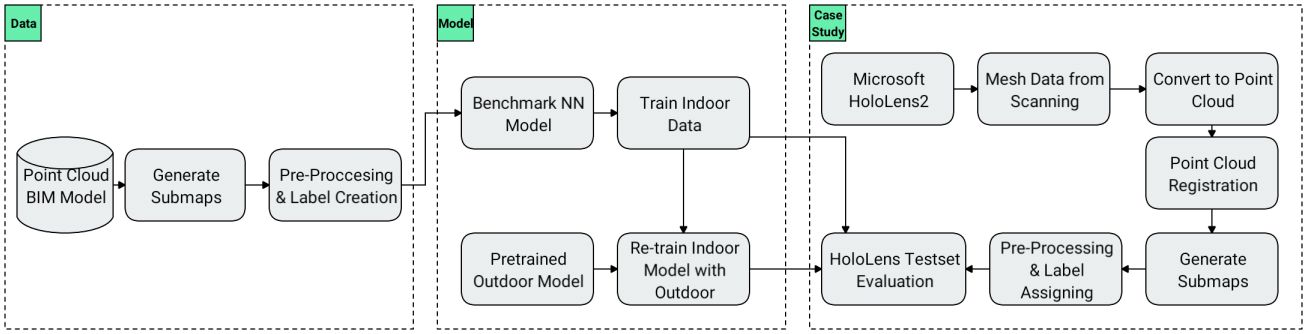


Figure 3.1: Our system framework includes three main phases; Data gathering, deep learning model, case study as an application.

4.1(b). The significance of this capability is that by walking and scanning The environment, dynamic conditions like the crowd or random visual elements which are not part of the main structure would be removed automatically. This process is done by scanning over time and updating the impurity of real-time data. After collecting the mesh models, 3D submaps are generated and pre-processed as same as the original laser data to be used as a testbed to see how effectively and accurately the deep models work for a complex indoor space. Furthermore, these models have been improved by the Transfer Learning method in which by transferring an outdoor localization model and retrained it with our indoor dataset we could improve the localization results.

For demonstrating our work in this research, a framework is shown in figure 3.1 which includes three main phases: (1) data; (2) Model; and (3) case study. The

data phase represents the collection of point cloud data with a stationary laser scanner and registering them to make our point cloud BIM model. Then our dataset was prepared by generating submaps with pre-processing that includes downsampling, normalization, and label assigning (refer to Chapter 4). The model phase represents the indoor localization deep neural network training, evaluation, and also improvements with Transfer Learning (refer to section 3.0.1). At last, we have our case study phase which represents the HoloLens2 testbed that has been collected in mesh scanning data by a person walks through an indoor environment. Then mesh is converted to point cloud to be registered and pre-processed in order to make the HoloLens testsets evaluated with the trained model (refer to Chapter 4). In the next section, we will go through the deep neural network models architectures and metric learning to see how one point cloud input gets processed and translated into a global descriptor vector. Then we propose a semi-real-time system that developed on a HoloLens UWP application that connects to a local server to localize itself by sending queries to the model.

3.0.1 Benchmark Models

The developed model in this study was inspired by multiple state-of-the-art neural network models which were initially designed for 3D point cloud object detection and large-scale place recognition in outdoor environments. These mod-

els apply the LIDAR (Light Detection And Ranging) scanners dataset to train their models as a trajectory-based approach for driving cars outside. Building on existing work, we trained our models to identify user locations given the spatial conditions. The problem within indoor environments is more complex than outdoor conditions as there are much fewer features points that we can train our models due to the smaller sizes of the submaps generated which result in less information each submap encapsulates. In addition, the structural differences in various places in an indoor environment are very diverse.

***PointNet*:** Qi et al. [27] introduced PointNet which consumes unordered 3D point cloud raw points as inputs instead of usually transferred them into a 3D voxel grid or images. Each point is described by its coordinates (x, y, z) and other features like color (RGB) values or normal values. However, PointNet and other inspired models reflected in this research only rely on the three coordinates as a point representation. Qi et al. indicated that their key approach is using a single symmetric function called max pooling. To achieve this, the system selects the maximum value for each patch of the entire feature map to detect the most important aspects of the local changes by aggregating the local point features which are invariant to the small translations and noises. After the max-pooling layer, a fully connected layer is designed to convert the optimal values into the

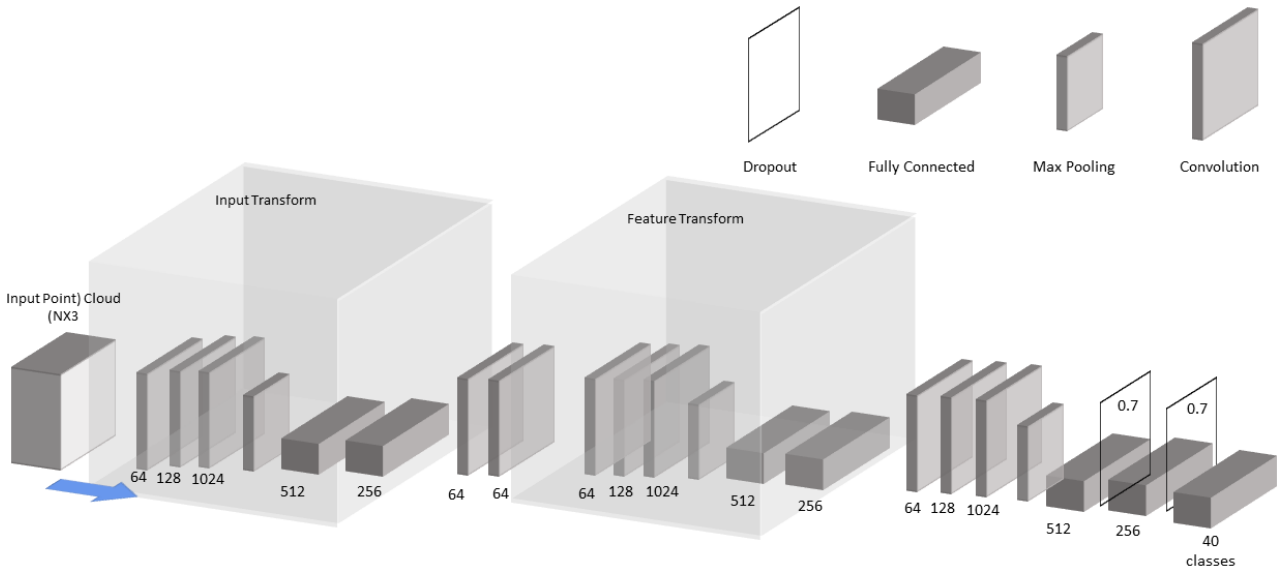


Figure 3.2: PointNet Architecture.

global descriptor for classification or segmentation problems. Figure3.2. demonstrates the PointNet network architecture.

***NetVLAD*:** The NetVLAD [4] model proposes before pointNet which led into 3D point cloud based place recognition advances in computer vision. The most significant purpose of this work was using two convolutional neural network base architectures including AlexNet [22] and VGG-16 [30] (cropped before the last convolution+ReLU [26] layer) to aggregate the local descriptors from the entire input 2d image and compress them into a single global descriptor vector (VLAD core which is inspired by [20, 5]). The NetVLAD architecture is shown in figure 3.3. [20] stated that VLAD core is the aggregation of the difference between

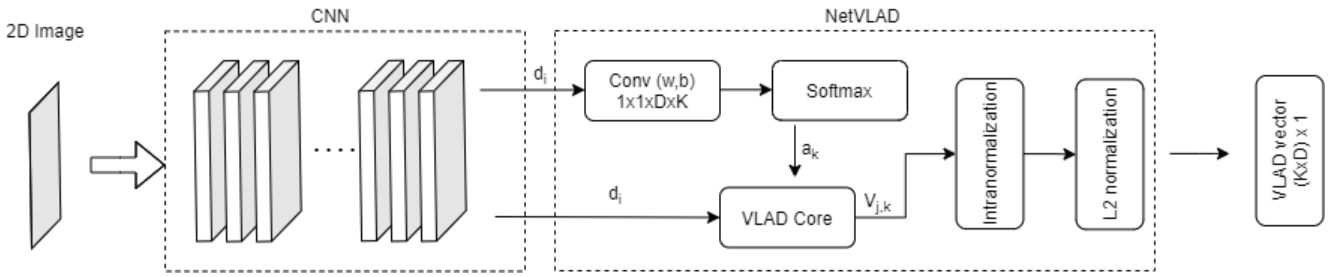


Figure 3.3: NetVLAD Architecture.

descriptor d and its corresponding visual word c_k for k clusters centers. The output of VLAD is represented by V as follows:

$$V(j,k) = \sum_{i=1}^{N} a_k(d_i)(d_i(j) - c_k(j)), \tag{3.1}$$

In the equation (1), we have a_k which is an indicator of a descriptor d_i association to a cluster center c_k for which value one is the closest association and value zero is the farthest. [4] showed that for using the VLAD core in an end-to-end place recognition, this layer should be differentiable in back-propagation operation. Therefore, instead of using the hard assignment a_k as zero or one, [4] introduced a replacement of a_k with a soft assignment of descriptors to clusters which would be differentiable with respect to all parameters and the input:

$$\bar{a}_k(D_i) = \frac{\exp\left(-\alpha \left\|d_i - c_k\right\|^2\right)}{\sum_{k'} \exp\left(-\alpha \left\|d_i - c_{k'}\right\|^2\right)} \tag{3.2}$$

Equation (2) indicates assigning the weights in correspondent with the distance

to the cluster centers which could be between zero and one unlike the rigid a_i formula proposed by [20]. Unless by putting α to $+\infty$, the original VLAD is reproduced. After simplifying the equation (2) and insert it into the original VLAD core equation (1), the following results are identified:

$$V(j,K) = \sum_{i=1}^{N} \frac{\exp\left(2\alpha d_i c_k - \alpha c_k^2\right)}{\sum_{k'} \exp\left(2\alpha d_i c_{k'} - \alpha c_{k'}^2\right)} (d_i(j) - c_k(j)) \tag{3.3}$$

$\{2\alpha c_k\}$ is defined by the w_k, $\{-\alpha c_k^2\}$, b_k, and c_k parameters which are trained throughout the network, whereas the original VLAD core has c_k as the only parameter to be trained. According to figure 3.3, we can see the CNN layers produce the output by training the $\{w_k\}$ and $\{b_k\}$ parameters with the $\{d_i\}$ input descriptors: $y_k(d_i) = w_k^T x_i + b_k$, then this CNN output goes through the softmax normalized exponential function $\sigma_k(Z) = \frac{\exp(z_k)}{\sum_{k'} \exp(z_{k'})}$. After the softmax level, we have soft assignment result $\bar{a}_k(D_i)$ which eventually is embedded into the VLAD core. Finally, there is an extra intra-normalization step iteallaboutvlad introduced to address a common issue in the context of image search, called burstiness [19]. This phenomenon happens when there are several similar visual elements (e.g. windows on a building, images taken from the same view, similar patterns on walls, ceiling, or floor, etc.) in the same image or amongst different images. The repeating patterns in the images have significant influences on similarity measures between images in which the most important aspects of the images would

be overlooked. Therefore the VLAD original approach was using Signed Square Root (SSR) to L2-normalize the entire descriptor vector of an image. However, [5] used L2-normalization within each VLAD blocks which is sum of residuals $(d_i(j) - c_k(j))$ for each cluster k. This helps us alleviate the burst impact on the point cloud descriptors where there are many repeating patterns like indoor walls with similar features.

3.0.1.1 PointNetVLAD

[32] proposed the point cloud-based retrieval for large-scale place recognition. The novelty of this work comes from combining the two aforementioned models that we discussed earlier to reach their goals. So far, we have learned that PointNet[27] takes 3D point cloud as input and after extracting the D-dimensional local feature descriptors following the max-pooling layer, it classifies with labels per submap or labels for each point (segmentation). Uy and et al. [32] connects the labels to the NetVLAD network for extracting global feature descriptors from the point cloud instead of the 2d image. According to the figure 3.4, after the fusion, a fully connected layer is added to decrease the vector with a high dimension (D x K). This reduction has been done for mitigating the computation complexity in order to be used for k nearest neighbor search to find the most similar and non-similar submaps in a training tuple. These descriptor vectors will be used for finding the best match for which the distance function $d(.)$, the Euclidean

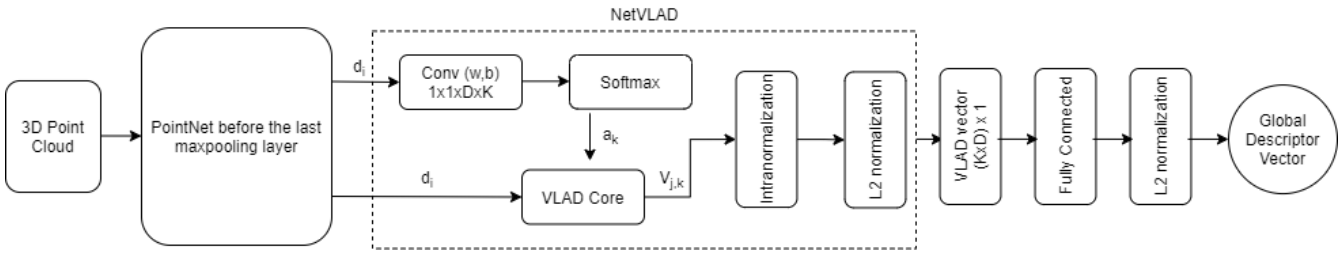


Figure 3.4: PointNetVLAD Architecture.

distance, is the smallest amongst other vectors.

***Metric Learning*:** PointNetVLAD proposed a function $f(.)$ which was trained in an end-to-end network to map a 3d point cloud input to a vector of global descriptors. In this research, we created training tuples similar to [32] in which we have a tuple for each input point cloud as $T = (P_a, P_p, \{P_n\})$. P_a is the anchor point cloud which is the focused point cloud as the center of the tuple, then we have P_p as a positive point cloud that has similarity with the anchor, and there are a subset of negative point clouds $\{P_n\}$ which are dissimilar to the anchor. This work has been done by introducing a modification version of the two well-known loss functions called the triplet loss and the quadruplet loss. Triplet loss which [29] introduced focuses on how we can ensure that an input image is closer to all images with same context than it is to the different images. This optimization tries to minimize the following statement:

min

$$\sum_{i}^{N} \Big[\|f(p_i^a) - f(p_i^p)\|_2^2 - \|f(p_i^a) - f(p_i^n)\|_2^2 + \alpha \Big]_+ \tag{3.4}$$

s.t.

$$\|p_i^a - p_i^p\|_2^2 + \alpha < \|p_i^a - p_i^n\|_2^2, \forall (p_i^a, p_i^p, p_i^n) \in T \tag{3.5}$$

In the equation (4) and (5), we have α which is a margin that is a determining factor between positives and negative pairs. This optimization would result in many triplets that are eligible due to the constraint. Therefore it would not be helpful in the training process because of the vast amount of triplets which causes a slow convergence. So instead of using these many triplets, hard negatives (closest negatives to the anchor) were used by removing the margin α from the equation (5) and decrease the training to the mini-batches that contain hard positive/negative samples from any anchor in the dataset.

[9] proposed the quadroplet loss that not only considers the relative distances between positive and negative images within the concentrated batch in training, but also by considering the new constraint in which it maximizes the additional distance between the randomly sampled negative image from the dataset that is dissimilar to the tuple T and the hardest negative in the tuple:

$$\sum_{i}^{N} \Big[\|f(p_i^a) - f(p_i^p)\|_2^2 - \|f(p_i^a) - f(p_i^n)\|_2^2 + \alpha \Big]_+ \tag{3.6}$$

$$+ \sum_{i}^{N} \Big[\|f(p_i^a) - f(p_i^p)\|_2^2 - \|f(p_i^n) - f(p_i^{n*})\|_2^2 + \beta \Big]_+ \tag{3.7}$$

[32] created the "Lazy" versions of these both losses in which it minimize the distance between the global descriptor vector of anchor and the positive submap $\delta_p = d(f(P_a), f(P_p))$, and maximize the distance between the negatives submaps and the anchor descriptor vector $\delta_{n_i} = d(f(P_a), f(P_{n_i} \in \{P_n\})$ by calculating the squared Euclidean distance. The term "Lazy" comes from switching the summations to the max operation which automatically detects the hard negative within the tuple.

$$LT = argmax_{n_i}([\delta_p - \delta_{n_i} + \alpha]_+) \tag{3.8}$$

This applies to the quadruplet as well in which the second term is finding the hard negative outside of the tuple and getting their distance to prevent undesirable reduction of distance between the hard negative in the tuple and another dissimilar point cloud outside of the tuple.

$$LQ = argmax_{n_i}([\delta_p - \delta_{n_i} + \alpha]_+) \tag{3.9}$$

$$+ argmax_{n_*}([\delta_p - \delta_{n_*} + \beta]_+) \tag{3.10}$$

In our models, we used lazy quadruplet loss as our metric learning same as [32].

3.0.2 Semi Real-Time System

To accomplish the semi real-time localization model on the HoloLens 2, we designed and implemented a query-based system which is depicted in figure 3.5 sequence diagram. This system is developed based on Model-View-Controller (MVC) software design pattern [7] that divides the program into three interconnected components. As shown in figure 3.5, the three modules Deep Network Model, Pre-processing in MATLAB, and PCD Edit and Conversion represent the Model component of the architecture. Then we have the control program and RESRful service API represents the Controller component of the architecture. This module takes the role of the central part of the program which controls the connections and data transfer between the user interface (HoloLens) and the server. At last, we have a UWP (Universal Windows Platform) application in the HoloLens, designed in the Unity software, which represents the View component of the architecture. Figure3.5 starts with the user's request for starting the scan by the HoloLens' augmented hologram. Then at the same time, the control program is always running and watching for an update on either the user side or the server side. Therefore, after detecting a request in the UWP app, the control pro-

gram fetches the .obj scan file via RESTful API and sends it to the pre-processing segment. In the pre-processing segment, a .obj which contains the scanned mesh data converts to a 3D point cloud and gets sent to the MATLAB code in order to get downsampled and normalized. Then the submap query gets sent to the deep neural network for getting the best-matched submaps coordinates. After that, the control program takes the results from the model and sends them to the UWP app. In the final state, the UWP app is in waiting mode to detect any uploaded results in the HoloLens file storage. After obtaining the results, they can be visualized as augmented information on the AR screen. This semi real-time system has been tested in which it takes around 10-15 seconds for the whole process to be completed and the spatial information such as sensors data visualized on the user's AR screen.

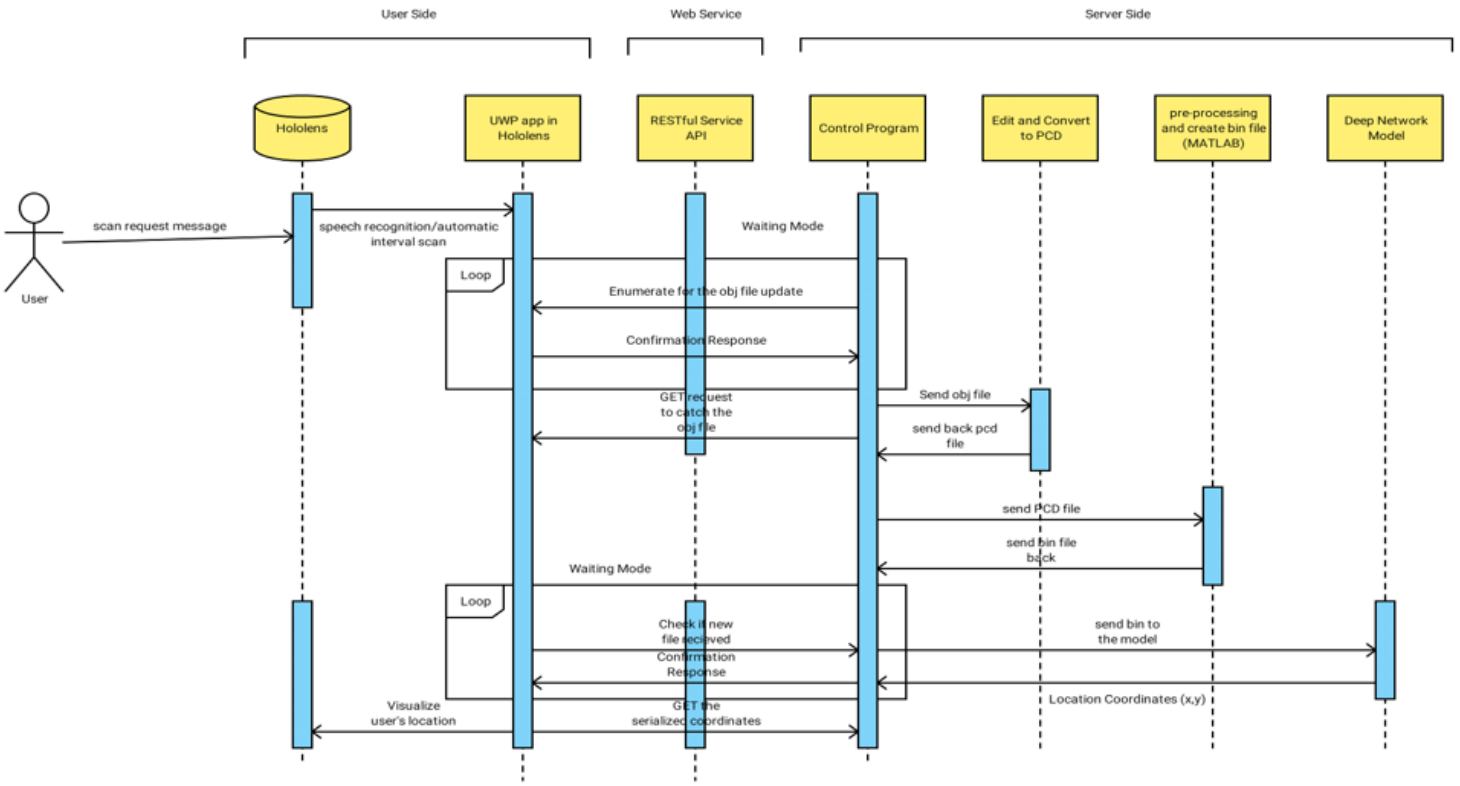


Figure 3.5: Semi real-time system schema is a sequence diagram of every process that a query goes through.

Chapter 4

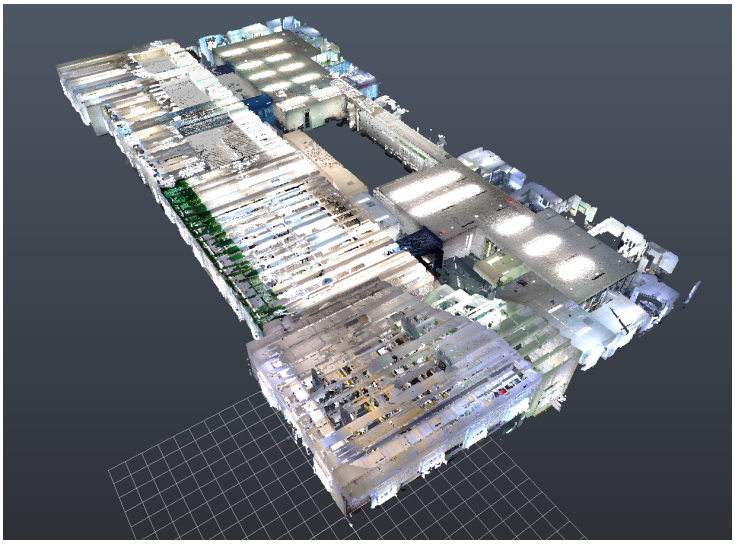
Data Collection and Experiments

We made an indoor point cloud model which was collected by two Faro Focus stationary laser scanners. This type of laser scanner is able to capture an accurate point cloud with less than a 6mm error rate up to 350m range. Additionally, Faro Focus utilizes advanced features in image capturing in different light and temperature conditions. For this study, we have collected and registered a 3D point cloud model with more than 194.1 million points from a 17,000 sq. ft. open office space which is located on the second floor of Olsson Hall at the University of Virginia campus figure 4.1(a). Also for our case study, we collected the mesh scans from a person wearing the Microsoft HoloLens2 and walk through an indoor environment. After that, we converted the mesh data into point clouds before registering them together in order to make the final 3D point cloud BIM model.

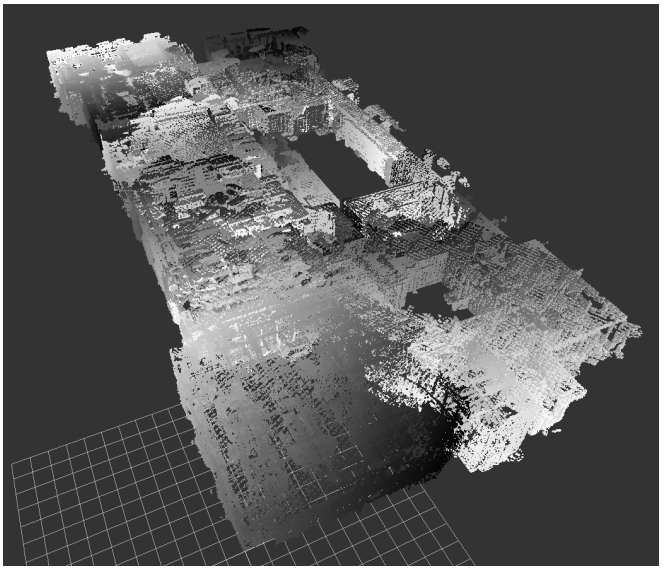
This map is shown in figure 4.1(b) which is made of more than 1.7 million points.

***Submap Generation*:** After registering the point clouds, we will have the final maps as shown in the figure 4.1. In previous work such as [32], each submap is generated based on the overlapped or disjoint frames that LIDAR collected over time for each run. However, in this work, we consider using the whole building point cloud, corner to corner, to make our submaps instead of just using trajectories taken by LIDARs. In order to accomplish this, we have assigned one of the corners of the building to the origin point (0,0) in order to have a global reference map as ground truth. Then we chose a 1mX1m regional margin from one of the corners and randomly selected a starting centroid point to crop the point cloud in fixed-sized regions with their centroids as labels. We repeat this process multiple times to make the dataset cover all areas with more than 40,000 submaps.

***Submap Preprocessing*:** After generating submaps from the 3D point cloud maps, they should be pre-processed to be ready for use as inputs to our neural network models. For the first part, the ceiling and floor were removed by a voxel grid filter in each submaps and for the second part, the ceiling and floor were part of them to see how much this information could affect the accuracy of the results which whether might play as a noise factor in the learning process or help



a) Laser scanner 3D point cloud map



b) HoloLens2 3D point cloud map

Figure 4.1: Our 3D point cloud referenced maps. (a) is the laser scanner-collected map and (b) is the HoloLens2-collected map as our case study

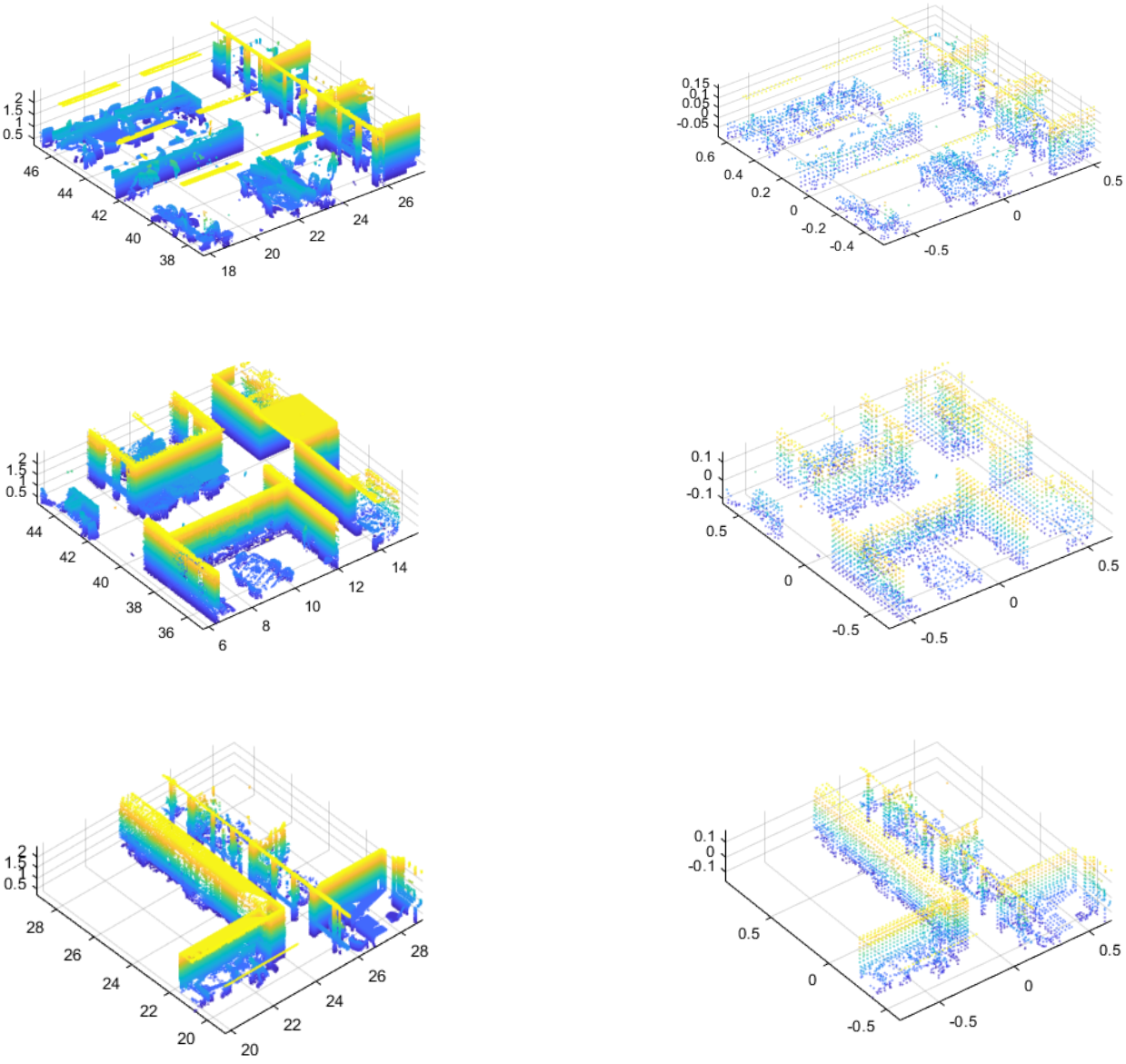


Figure 4.2: Three sample submaps from our dataset show the level of complexity and differences in structural architecture. The right side shows their pre-processed submaps

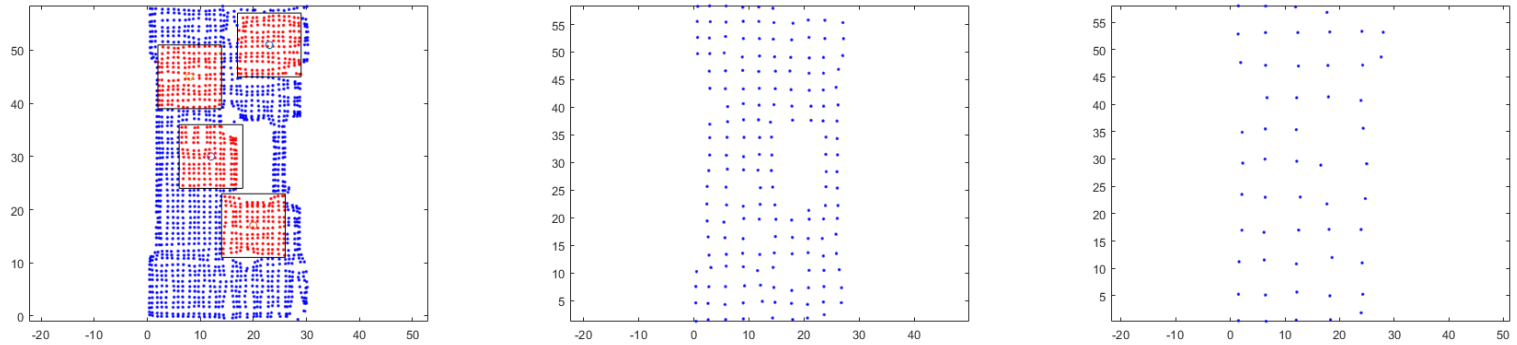
Table 4.1: The number of submaps distributions between training and testing in different input submap sizes.

		Training	**Testing**
	Input Size	***BL and TR***	***BL and TR***
Laser Scanner	3mX3m	37720	-
	6mX6m	37927	-
HoloLens2	3mX3m	-	4528
	6mX6m	-	1294

to enhance the localization. By optimizing the scale value regarding the number of points in each submap, we could come up with a fast downsampling rate to significantly reduce iterations to converge to the 4096 points. After downsampling, each submap normalized with zero mean with having values between -1 and 1 in order to be on the same scale for the deep neural networks. You could see a sample of the submap pre-processing in figures 4.2 that depicted the dynamic nature of an inside space such as noisy objects (office desks and chairs), crowd, narrow hallways, etc. After downsampling, new average values of axes (x, y) are calculated to be set as new submaps labels.

***Dataset Splitting and Evaluation*:** Laser scanner dataset that has 20 subsets

of joint (overlapped) submaps (each created with a random starting point with different labels), is split into training reference map and testing reference map as shown in figure 4.3(a). Then the laser dataset is used for training the indoor localization model. Whilst the model is being trained, evaluation loss is calculated with the specified random regions data points shown in figure 4.3(a) every 200 batches in each epoch. Then after the training has been done, we evaluated the trained model with the case study HoloLens2 disjoint (non-overlapped) testset in two different sizes shown in 4.3(b)and(c) which have been created in 20 runs with different labels. One of the sample test subset of each size is shown in figures 4.3(b) and (c). Table 4.1 shows the exact distribution of the training and testing submaps from the referenced maps. However, because of using various sizes of submaps, the number of disjoint submaps are different in testsets. The laser scanner referenced map has been cropped with intervals of 1 meter between each submap for trainsets. But HoloLens2 referenced map has been cropped in 3 and 6 meters intervals between submaps for testsets of sizes 3mX3m and 6mX6m respectively in order to not having any overlapped submaps within each run.



(a) Laser scanner Joint trainset (b) HoloLens2 disjoint 3mX3m testset (c) HoloLens2 disjoint 6mX6m testset

Figure 4.3: Data splitting: (a) represents a sample subset of the joint submaps of laser scanner trainset with 1-meter interval in which 4 regions were randomly selected for validating whilst training. (b) represents a sample subset of the disjoint submaps of HoloLens2 testset in 3-meter intervals. (c) represents a sample subset of the disjoint submaps of HoloLens2 testset in 6-meter interval

s

Chapter 5

Results and Discussion

In the results, we show that *PN-VLAD*, a large-scale point cloud-based place recognition which initially was designed for localizing a vehicle equipped with a LIDAR and camera system is applicable for indoor environments with various structural and architectural design, crowd sizes, and objects (furniture). In this section, we discuss the results of the proposed indoor localization method by testing the submaps generated from the HoloLens mesh/point cloud testset and report the average recall values of the top 1% results. Also, by the Transfer Learning approach, we improved our results significantly and found out that retraining outdoor and indoor localization models mutually could be useful to increase the accuracy. Also, we checked whether adding more data points for each submap such as floor and ceiling can be useful in results improvement.

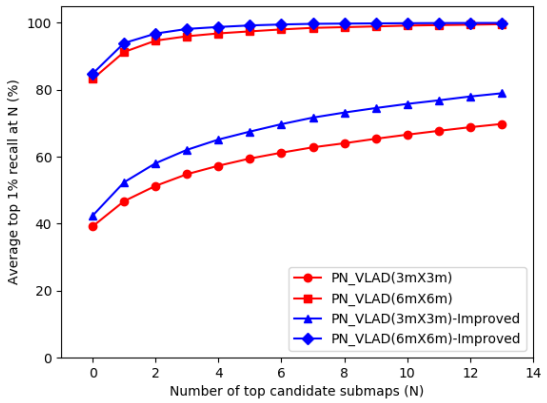
Table 5.1: Average top 1% recall(%) in baseline and transferred models tested on HoloLens testset in the threshold range of 1 meter.

	HoloLens	
trained model	***Baseline***	***Transferred***
PN-VLAD (3mX3m)	39.18	52.08
PN-VLAD (6mX6m)	83.15	**90.59**

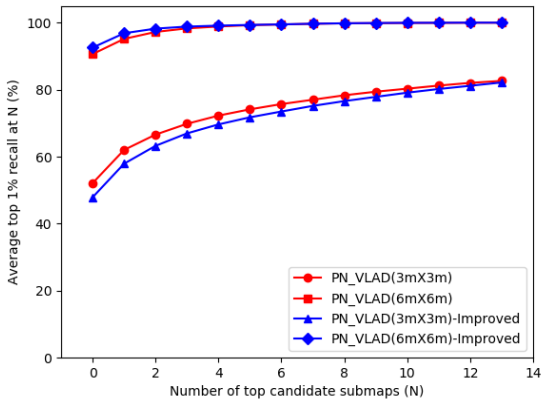
Networks and Input Size Variations The laser scanner data was trained by *PN-VLAD* to build the baseline models (without floor and ceiling). The models are trained with the same settings to produce a 256-dim global feature vector to represent a submap. Due to the fixed-size vector, any raw submaps with distinct input sizes do not make difference in computation speed or power. Models are trained with quadruplet loss with the margins $\alpha = 0.5$ and $\beta = 0.2$ alongside Adam [21] optimizer. Number of cluster is set to be $k = 64$. We used batch size of 2 tuples for each iteration in training that has one anchor input submap p_a for which we have 18 negatives and 2 positive submaps as same as [32]. We compare the performance of our baseline model (trained models from the laser

Figure 5.1: Average recall of the top submap candidates resulted from the HoloLens testset data. The blue charts show the models with floor and ceiling (improvement) in two sizes. The red charts show the models without floor and ceiling.

(a) Baseline Network



(c) Transferred Network



scanner indoor data) with the *transferred* model in two input sizes in testing on the HoloLens2 testset. *Transferred* states the Transfer Learning method in which we would transfer a trained model with the same task but in a different domain (or vice versa) which in this case we used the trained localization model on an outdoor dataset Oxford [34] and retrained it with our indoor laser scanner dataset. According to the table 5.1, transferred model improved the top 1% recall rate from the baseline model in both input submap sizes 3X3 and 6X6 meters. Also, figure 5.1 shows the average recall values of the top 14 candidates in which we can see the transferred networks in both sizes outperformed better in terms of the first starting value in comparison with the baseline networks. The transferred model in the 6m X 6m submap dataset receives the highest recall rate by 90.59% via the testing of our HoloLens case study.

Models improvement To see if we can improve our models, we generated the submaps without removing the ceiling and floor (unlike the last section models which did not include these features). As shown in table 5.2, both input submap sizes improved in each model for each input size tested on the HoloLens testset which means adding static information from the environment to the submaps could help discriminate between them and be recognized more accurately. Figure 5.1 is also shown the difference in which the improved models with floor and

Table 5.2: Average top 1% recall (%) in baseline and transfer models tested on HoloLens testset with floor and ceiling in the threshold range of 1 meter.

	HoloLens	
trained model	***Baseline***	***Transferred***
PN-VLAD (3mX3m)	52.35	57.88
PN-VLAD (6mX6m)	84.74	**92.53**

ceiling (blue charts) show the improvement from the models without floor and ceiling (red charts) in both sizes. These results imply that in some indoor spaces that we have limitations in scanning a wide area due to occlusions or being in small space, more information from the environment such as floor and ceiling or structural elements and static objects could enhance differentiating between the generated submaps.

Comparison between various areas in our indoor space We also studied the baseline and transferred models in different areas within our space as shown in table 5.3 via the HoloLens test dataset in different submap sizes. According to table 5.3, our models performed with high recall rates in all regions as illustrated

Table 5.3: Average top 1% recall (%) in baseline and transferred models tested on the HoloLens test data with floor and ceiling in various indoor areas.

Submap Size	**(3mX3m)**		**(6mX6m)**	
trained model	***Baseline***	***Transferred***	***Baseline***	***Transferred***
Student Seating Areas	43.94	52.73	72.65	86.78
Hallways	43.21	50.32	84.25	89.90
Long Corridor	52.41	66.00	88.68	100.0
Cafeteria(Open Arena)	56.59	53.91	96.00	96.03
Hardware Lab	54.46	60.86	88.35	94.70
Conference Rooms	67.46	68.53	73.94	96.09

in figure 5.2. If we reduce the range of scanning for the 3x3 size of the input, the results in both baseline and transferred models would decrease accordingly for every indoor area in comparison with 6x6 scan size. For example hallways (P8-P14 in the figure 5.2) and student seating areas (P1-P7 in the figure 5.2) have the least amount of recalls amongst others in size 3 which shows in small and crowded areas, the less information/points can affect the localization negatively. Furthermore, we can see that student seating areas (P1-P7 in the figure 5.2) have the least recall rates by 86.78% in comparison to others. As it is depicted in figure 5.2, in the seating areas, there are much more dynamic objects such as chairs and

desks arrangement than other sections which can be challenging because dynamic environments change over time.

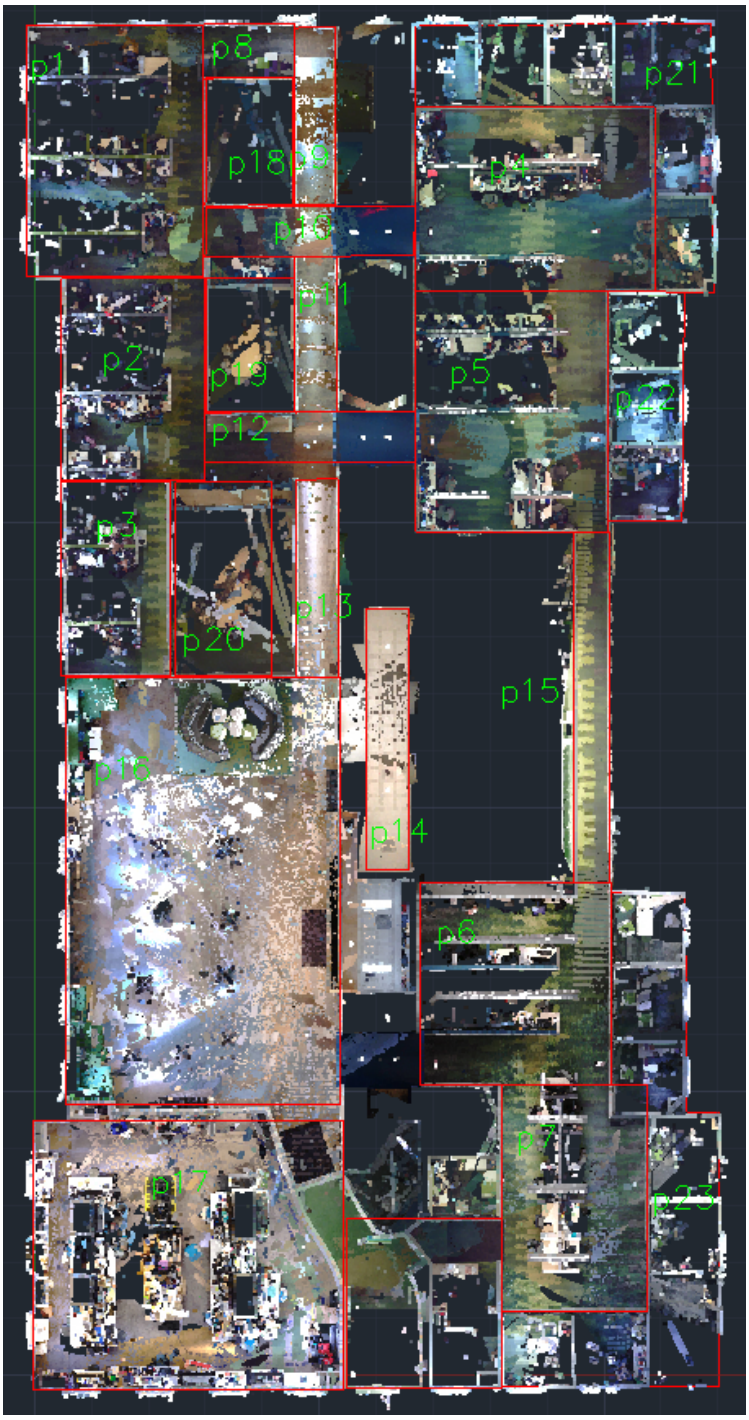


Figure 5.2: Various sections of the Link Lab. P1-P7: student seating areas, P8-P14: hallways, P15: Long Corridor, P16: cafeteria(Open Arena), P17: hardware lab, P18-P20: conference rooms, P21-P23: single office rooms.

Chapter 6

Conclusion, Limitations, and Future Works

In this work, we introduced an AR-based indoor localization system to be used in applications such as facility management, construction management, and emergency response. For accomplishing this purpose, we utilized laser scanners to create a 3D point cloud of an indoor environment. Then as a visual place recognition technique, a deep learning-based model *PointNetVLAD [32]* was used to train different models to see what would give the best results according to the given conditions. Furthermore, we improved the initial results both in the laser scanner and our case study (on HoloLens 2) via transfer learning in which the outdoor localization trained model was transferred and retrain with our dataset.

By including additional information from static elements such as ceiling and floor, we were able to improve the performance of each of our models to more than 90% recall rate of the average top 1% of the results in a 1-meter threshold. Finally, we studied various areas in our case study environment such as hallways, conference rooms, or student seating area that includes a lot of dynamic objects and dense crowds. Results showed that our baseline and improved models were successful and consistent with the overall results of 85.20% and 93.75% for baseline and transferred models respectively.

One of the limitations of this work is the size of our dataset. In compare the Oxford Robotcar dataset, our dataset is small scale because of its closed-environment characteristic which limits us in the number of disjoint submaps. As a future step to address this limitation, the model should be tested in more indoor environments to see the generalizability of the proposed method. Additionally, one of the laser scanner drawbacks is the laser reflection on the glass doors and walls which can affect the point cloud quality around that area. For instance, in our dataset, the single-occupancy offices shown in P21-P23 in figure 5.2 are made of glass partition which the laser scanner does not provide an accurate map of the space, and therefore, it is impossible for us to evaluate the model inside these offices. Lastly, Simultaneous localization and mapping (SLAM) methods [28, 23], commonly used for indoor localization of robotic systems in indoor environments,

need to be evaluated and compared to the proposed PointNetVLAD method in the context of our approach.

For recognizing the user's direction and orientation, in our future works, we will be considering using HoloLens' capability of gaze tracking and its IMU sensors as other useful information that could be added to the proposed framework for better localization and orientation of the user. Additionally, in our future work, we will integrate the user intention regarding their movement and gaze direction so the situational awareness system could visualize some information in the AR platform to help towards making decisions. Lastly, one limitation of HoloLens is its computational capability, resulting in near-real-time (10-15 seconds) processing of information in our designed system. This limitation can be addressed if the processing is moved from the device (HoloLens) and our local server to the Microsoft Azure cloud services which connects to the HoloLens directly and does all the computation including submap processing and deep neural models.

Bibliography

[1] D. Acharya, K. Khoshelham, and S. Winter. Bim-posenet: Indoor camera localisation using a 3d indoor model and deep learning from synthetic images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 150:245–258, 2019.

[2] A. Ahmad, P. Claudio, A. Alizadeh Naeini, and G. Sohn. Wi-fi rss fingerprinting for indoor localization using augmented reality. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:57–64, 2020.

[3] B. Akinci. Situational awareness in construction and facility management. *Frontiers of Engineering Management*, 1:283, 2014.

[4] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, page 5297–5307, 2016.

[5] Relja Arandjelovic and Andrew Zisserman. All about vlad. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, page 1578–1585, 2013.

[6] F. Baek, I. Ha, and H. Kim. Augmented reality system for facility management using image-based indoor localization. *Automation in Construction*, 99:18–26, 2019.

[7] J. Bucanek. Model-view-controller pattern. in learn objective-c for java developers. *Apress*, 2009.

[8] Haosen Chen, Lei Hou, Guomin (Kevin) Zhang, and Sungkon Moon. Development of bim, iot and ar/vr technologies for fire safety and upskilling. *Automation in Construction*, 125, 2021.

[9] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: A deep quadruplet network for person re-identification. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 403–412, July 2017.

[10] B. A. Delail, L. Weruaga, and M. J. Zemerly. Caviar: Context aware visual indoor augmented reality for a university campus. *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, page 286–290, 2012.

[11] R. Diao, V. Vittal, and N. Logic. Design of a real-time security assessment tool for situational awareness enhancement in modern power systems. *IEEE Transactions on Power Systems*, 25:957–965, July 2010.

[12] F. T. Durso and R. S. Nickerson. In handbook of applied cognition. *J. Wiley*, page 283–314, 1999.

[13] MR Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995.

[14] M.R. Endsley. Supporting situation awareness in aviation systems. *IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, 5:4177–4181, 1997.

[15] J. Giri, M. Parashar, J. Trehern, and V. Madani. The situation room: Control center analytics for enhanced situational awareness. *IEEE Power and Energy Magazine*, 10:24–39, 2012.

[16] F. Höflinger, R. Zhang, and L. M. Reindl. Indoor-localization system using a micro-inertial measurement unit (imu). *European Frequency and Time Forum*, page 443–447, 2012.

[17] P. Hübner, M. Weinmann, and S. Wursthorn. Marker-based localization of the microsoft hololens in building models. *The International Archives of the*

Photogrammetry, Remote Sensing and Spatial Information Sciences, page 195–202, 2018.

[18] J. Irizarry, M. Gheisari, G. Williams, and B.N. Walker. A mobile augmented reality method for accessing building information through a situation awareness approach. *Automation in Construction*, 33:11–23, 2013.

[19] H. Jegou, M. Douze, and C. Schmid. On the burstiness of visual elements. *IEEE Conference on Computer Vision and Pattern Recognition*, page 1169–1176, 2009.

[20] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 3304–3311, 2010.

[21] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980 [cs]*, 2017.

[22] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60:84–90, 2017.

[23] S. Ryu M. Y. Chang, S. Yeon and D. Lee. Spoxelnet: Spherical voxel-based deep place recognition for 3d point clouds of crowded indoor spaces. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8564–8570, 2020.

[24] C. Masone and B. Caputo. A survey on deep visual place recognition. *IEEE*, page 19516–19547, 2021.

[25] A. Morar, A. Moldoveanu, I. Mocanu, F. Moldoveanu, I. E. Radoi, V. Asavei, and A. Gradinaru. A comprehensive survey of indoor localization methods based on computer vision. *Sensors*, 20(9):2641, 2020.

[26] Hinton G.E. Nair, V. Rectified linear units improve restricted boltzmann machines.

[27] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, page 652–660, July 2017.

[28] P. Wu S. Chan and L. Fu. Robust 2d indoor localization through laser slam and visual slam fusion. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1263–1268, 2018.

[29] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 815–823, 2015.

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *In Proc. ICLR*, 2015.

[31] S. Taneja, A. Akcamete, B. Akinci, J.H. Garrett, L. Soibelman, and E.W. East. Analysis of three indoor localization technologies for supporting operations and maintenance field tasks. *Computing in Civil Engineering*, 26:708–719, 2012.

[32] Mikaela Angelina Uy and Gim Hee Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, page 4470–4479, June 2018.

[33] M. Vidulich, c. Dominguez, E Vogel, and G. McMillan. Situation awareness: Papers and annotated bibliography. *Armstrong Laboratory Technical Report, AL/CFTR-1994-0085.*, 1994.

[34] C. Linegar W. Maddern, G. Pascoe and P. Newman. 1 year, 1000km: The oxford robotcar dataset. *The International Journal of Robotics Research (IJRR)*, 2017.

[35] C. Wang, W. Su, and Y. Guo. An augmented reality mobile navigation system supporting ibeacon assisted location-aware service. *International Conference on Applied System Innovation (ICASI)*, page 1–4, 2016.

[36] C. S. Wang, C. Ding-Jung, and H. Yi-Yun. 3d augmented reality mobile navigation system supporting indoor positioning function. *ISPRS Annals*

of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2012.

[37] Y-T Wang, C-C Peng, A. A. Ravankar, and Ravankar A. A single lidar-based feature fusion indoor localization algorithm. *Sensors*, 2018.

[38] Y. Wei and B. Akinci. A vision and learning-based indoor localization and semantic mapping framework for facility operations and management. *Automation in Construction*, 107, 2019.

[39] Han Xu, Z. Yang, Z. Zhou, L. Shangguan, K. Yi, and Y. Liu. Indoor localization via multi-modal sensing on smartphones. *Association for Computing Machinery*, page 208–219, 2016.

[40] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios. Recent advances in indoor localization: A survey on theoretical approaches and applications. *IEEE Communications Surveys Tutorials*, 19:1327–1346, 2017.

[41] F. Zafari, A. Gkelias, and K. K. Leung. A survey of indoor localization systems and technologies. *IEEE Communications Surveys Tutorials*, 21:2568–2599, 2019.

Appendix A

Guideline for Future

We provide a guideline in which it's described what actions we took to build our indoor localization model for the future reference step by step as following:

1. We collected our 3D point cloud model with FARO Focus laser scanner.

Figure A.1: Faro Focus Laser Scanner in our dataset location.

Figure A.2: Sample view of the laser scanner point cloud data.

2. For our case study, we collected the test data with Microsoft HoloLens2 via a user walks through the indoor environment and collects the triangular meshes from around him/her. Then this mesh maps should be converted to a 3D point cloud.

Figure A.3: Microsoft HoloLens 2.

Figure A.4: Sample view of HoloLens2 spatial mesh data.

3. In the next step we registered the point cloud scan files with the Autodesk ReCap software to have an integrated and accurate 3D model. Here you can follow the guideline shows in Manual Registration for registering structured (fixed-location e.g. Faro laser scanner) point cloud scan files.

4. For registering the HoloLens2 point cloud files, you should follow the guideline in Registering Unstructured Scans. As a summary, for registering unstructured scans (photogrammetry, handheld, or mobile) which HoloLens is part of, we should combine them with the structured scan files to get them

registered then by removing structured data, finally, we have integrated unstructured 3D point cloud of HoloLens.

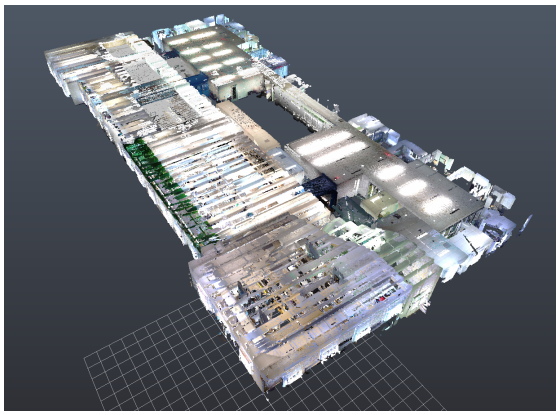


Figure A.5: Laser scanner 3D point cloud map.

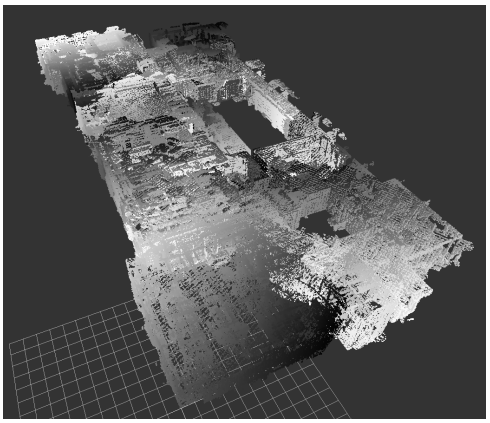


Figure A.6: HoloLens2 3D point cloud map.

5. In Recap, we assigned one of the model's corners as our origin point (0,0) to be used for reference on a global scale.

6. For generating submaps, we used Open3D which is a python library for processing 3D data. (refer to number 13, Generating Submaps folder in our GitHub repository)

7. In Open3D, we assign a region of 1m X 1m for randomly selecting a starting point in normal distribution for generating submaps for each run and each subset.(refer to number 13, Generating Submaps folder in our GitHub repository)

8. Laser scanner 3D point cloud map was used for training subsets in which the interval between submaps was set to 1 meter for both submap sizes 3 and 6 meters. HoloLens2 3D point cloud map used for testing in which intervals were set to 3 and 6 meters for submap sizes 3mX3m and 6mX6m respectively in order to generate disjoint submaps without overlapping areas in each testing subset. (refer to number 13, Generating Submaps folder in our GitHub repository)

9. After generating submaps, they have to be pre-processed to be entered into the deep network. Therefore, by using the MATLAB point cloud downsampling function we can downsample each point cloud to 4096 points. Then each of them is normalized with zero mean and new (x,y,z) values between -1 and 1. Also, new labels based on the new axes values are calculated and assigned to each submaps. (refer to number 13, Submap pre-processing folder in our GitHub repository)

10. Next, we made our training tuples ($\{p_a, p_p, p_n\}$) consist of an anchor point, positives points, and negative points by using k-nearest neighbors for every submaps, at the same time we split the submaps into training and testing references by defining random regions of 6mX6m and saved them in pickle files. (refer to number 13, Generating Tuples/generate_training_tuples_baseline.py file in our GitHub repository)

11. Then we trained the baseline deep model by loading the laser scanner tuples in 10 epochs in which those testing regions were used for evaluation loss calculation in every 200 batches. (refer to number 13, Main.py file in our GitHub repository)

12. Then we generated the testing tuples ($\{p_a, p_p, p_n\}$) for evaluating the trained model (step 11). In this step, all the ground truths of queries are extracted from the whole testset except the subset that includes the targeted query in each iteration to avoid self-matching. This has been done by K-nearest neighbors and we set 1-meter as our positive range (k). (refer to number 13, Generating Tuples/generate_test_sets.py file in our GitHub repository)

13. Finally, we evaluated the trained model (step 11) with the HoloLens disjoint queries testsets to see how good the trained model localize the queries that come from a different device with a different way of collecting data which is the mesh collection by a user walked through the indoor space. The mesh data has to be converted to 3D point cloud data with the Open3D library. In this step, the top global descriptor vectors which are the model outcomes will be checked with the ground truths we calculated in the previous step to see how many of the top results exist in the ground truths of the query. (refer to number 13, evaluate.py file in our GitHub repository)

14. Our code can be found in our github repository. Also, all referenced maps

including a laser scanner and HoloLens2 3D point cloud BIM models with and without floor and ceiling can be found in our UVA box.