Abstract—Visual positioning is a critical function in applications such as navigation and extended reality experiences. Recently, deep learning technologies, especially classification, have been implemented on the positioning task. However, to acquire a comprehensive positioning dataset and produce a high-performance neural network model is challenging. In this article, we propose to solve the issue by projecting training images from auto-generated 3D point cloud maps. By utilizing branch convolutional neural network (B-CNN) model, the “zoom-in” equivalent property results in favorable positioning accuracy and successful real-time implementation.

I. INTRODUCTION

In recent years, position-based services, such as autonomous driving and indoor navigation, is rapidly gaining popularity. While outdoor positioning and navigation mainly use GPS-based methods, indoor devices receive almost no GPS signals requiring very different localization solutions. Furthermore, with the emerging of augmented reality and virtual reality (AR/VR) smart applications, accurate real-time indoor positioning becomes one of the core technologies.

With increasing computation power, various kinds of camera-equipped devices become primary sensors for visual place recognition. The conventional bag-of-words (BoW) model with an inverted index is efficient at retrieving similar images based on extracted features, but it is less capable of discovering fine-grained concepts in the query and returning semantically matched results [1]. As the image dataset grows and the number of interest positions becomes large, the complication is that visual separability between different places is highly uneven. Some categories are much harder to distinguish than others.

At the same time, machine learning approaches to visual-based positioning draw lots of attention. Most solutions form positioning as a regression problem, which requires a vast set of pictures to represent geographical locations and counts on the relative distance between input pictures and predefined locations to deduce the camera pose [2]. However, the cost of collecting images of all locations across a large coverage area with accurate coordinates is high and infeasible in many cases.

In this paper, we propose a novel convolutional neural network (CNN)-based scheme treating visual positioning as a classification problem for environments with 3D point clouds available. This system adopts the branch convolutional neural network (B-CNN) [3] for hierarchical positioning. The coarse-to-fine classification nature of B-CNN realizes the desired "zoom-in" effect to enhance the positioning accuracy progressively. Instead of collecting pictures at all possible locations, we propose a projection method to generate RGB and depth images from arbitrary viewing points in the 3D point clouds. The main contributions are to:

- Generate images from 3D environment maps to train positioning models allowing to lower the cost of acquiring training data significantly.
- Form the visual positioning as an image classification problem and take advantage of B-CNN to improve performance over fine-categories.
- Implement the process on smartphones to demonstrate the real-time positioning capability of the proposed system.

Overall, the trained CNN model is specialized in mapping RGB and depth images to a position in the environment. The distance and viewing angle can be determined at the same time. The experiments use a well-recognized 3D mapping tool, RTAB-Map [4], showing high validation accuracy during the training process, advantages of the coarse-to-fine classification model, and high final positioning accuracy.

The paper proceeds as follows. Section II describes related works on 3D positioning and place recognition. Section III explains the way to project 3D point clouds to 2D images and the neural network structure. Section IV presents the implementation and performance evaluation. Section V demonstrates the smartphone implementation. Conclusions are drawn in Section VI.

II. RELATED WORK

In recent years, several indoor positioning and loop closure detection papers have been published. Liu et al. established a survey of indoor positioning techniques and systems [5]. Ottersten et al. developed the AOA Estimation [6] and made progress on wireless positioning. However, the wireless signal usually interfered with obstacles and degraded the positioning accuracy. Thus visual-based positioning methods get increasing attention. Lowry et al. published a survey about visual place recognition [7], which is useful for loop closure detection. Angeli et al. [8] presented an online method which makes it possible to detect an image coming from a perceived scene using local shape and color information. Gehrig et al. [9] presented methods for both 2D-to-2D pose-graph vertex matching and 2D-to-3D landmark matching based on scoring. Labbe et al. [10] published a loop closure detection system for large-scale and long-term operation. This paper used memory...

Deep learning has developed rapidly, especially on object detection and image classification [13]. Indoor positioning methods using CNN is emerging as well. Handling the positioning as a regression task, authors in [14] trained a CNN to estimate camera position; the work in [15] combined CNN and LSTM to localize the input photo. Classification is another main problem in computer vision. CNN structures are designed to learn internal representations that abstract from the input data based on hierarchical layers. It is natural to ask if the concept can be applied to learn a model across hierarchical classes using multiple output layers. B-CNN [3] utilized the idea by outputting an extra coarse branch on CNN.

III. METHODOLOGY

The positioning system has two major parts. In the first part, we build 3D models of indoor environments through the open-source software RTAB-Map [4]. The 3D model is composed of point clouds with RGB color and depth information. After that, we project the point clouds from different viewing locations and angles to generate images. In the second part, we train the deep neural network models with generated images for positioning.

A. Training Image Generation

To generate training RGB and depth images, we project 3D point cloud map to 2D planes. The projection plane simulates the screen users can see. Since the goal is projecting cloud points onto the plane according to the user’s view, the relative coordination between the user, projection plane, and cloud points need to be evaluated according to the line of sight.

Next, we divide the 3D map into grid areas, where each grid area is a category to be classified. The point clouds are projected, assuming the user is at the center of each grid area. Therefore, the visual localization becomes a classification problem suitable to apply deep learning techniques. Because 3D-to-2D projection can generate images from arbitrary number views, there is no need to take plenty of extra photos for each location, which is a significant advantage of this approach. Fig. 1 shows the indoor environment of a National Central University (NCU) engineering building with predetermined grids, and the arrows are possible viewing directions from the center point. The actual projected images are also shown below.

Inevitably, the generated images are noisy comparing to realistic photos, as we can see missing pixels in Fig. 1. We compensate it though denoise pre-processing.

B. Convolutional Neural Network Architecture

To achieve “zoom-in” effect on the positioning problem, we adopt the B-CNN [3] concept. The B-CNN assigns additional coarse labels in the mid-stage of CNNs, so fine categories can be better classified. We apply this concept to the positioning problem. Fig. 2 shows that use the larger grids on the map as a coarse classification and smaller grids on the map as the final classification.

Due to perceptual aliasing, that is, two different places are perceived as the same, we use pair of photos as input data for deep learning models, each pair of photos has 30-degree viewing angle in the horizontal plane, for example, 0 and 30, 1 and 31, and then each class has 360 samples. If we want to produce more data for training and testing, cut the photo into four equal parts with some overlap. To overcome the model
learning problem, like internal covariant shift, and make sure that the different features take on similar ranges of values, we normalize the RGB photos and depth photos. An essential technology in our work is the fusion of RGB and depth photos in the deep neural network. There are six different input cases, including single RGB image, single depth image, single RGB and depth image, continuous two RGB images, continuous two depth images, and continuous two RGB and depth images. Each continuous image pair has 30 degrees interval and supply input data, with more features be detected. With this information, B-CNN can extract features from both RGB and depth images and improve positioning accuracy. Fig. 3 shows continuous two RGB and depth images as input in B-CNN architecture.

IV. EXPERIMENTAL RESULTS

We create 16 and 64 rectangular mesh grids in the National Central University (NCU) engineering building environment, and each grid has 1 meter and 0.5-meter interval respectively, see Fig. 1. The training data size is 224*224 pixels, and all the training data passed through the median filter for pre-processing. The size of B-CNN branches is determined empirically to strike the right balance between the ability to differentiate category and the size of the required network. The 16 grids case has one branch, and the 64 case has two branches with adaptive loss weights. The rule of adaptive loss weight changes according to the epoch of the training process. The proportion of branch loss weight is more significant than the final loss weight at the early training epoch. With the epoch grows, the proportion of final loss weight also raises, and it will take the full dominance at the two-thirds of epochs. We also compare with the VGG16 [16]model and testify six different input cases performance with two precision.

A. Training Analysis

To evaluate the accuracy of the purposed method in the real world, we exploited TurtleBot robot equipping with Kinect V1 to take images in the NCU engineering building. The testing images comprises 56 RGB and depth images, we discard some of the photos with few features in advance, such as pictures filled by the wall or no object appeared on it will be removed. Table I shows the validation accuracy after training progress in the NCU engineering building. As we concerned about the more abundant information input, the higher the accuracy it gets. The most significant advantage of inputting two continuous images is that when one single image input has a sparse feature, a neural network is difficult to recognize the view in the environment. Furthermore, we took the depth images into account; the distance information between feature and viewpoint also helped the model to improve the positioning accuracy.

B. Testing Analysis

Fig. 4 shows the positioning performance under various input combinations in the number of correctly recognized positions. The result shows that the case of continuous two RGB images and continuous two RGB and depth images have improved performance. The high number of repeated scenes and large single color decoration areas confuse the model. Besides, the indoor environment 3D map does not cover the whole place, such as the ceiling. The trend of input cases with depth images is better than others, which verifies the depth information has a positive influence. Fig. 5 are three successfully positioned samples when inputting continuous RGB and depth images. The 3D map generated training pictures are on the left, and testing pictures from a camera are on the right. In Fig. 5a, the training data has a similar composition to the testing picture, but with some defects on the bottom, the model can easily manage it. The next example in Fig. 5b suffers from reflection, distortion, and missing data; the generated image is profoundly different from the real scene. By combining multiple views and depth information, the query images can still be classified into the correct position. In Fig. 5c, the distortion makes the model position incorrectly, but the error is limited within one grid.

V. IMPLEMENTATION

We implement the purposed system as an iOS application called Camera Positioning System (CPS). Users can take photos in the environment. By providing two continuous photos around, the application will point out the user’s position on the map. Fig. 6 shows the positioning progress. Two photos on the left screenshot are inputs; the right screenshot outputs the location where the photos are taken.

VI. CONCLUSIONS

In this paper, a visual-based indoor positioning with a "zoom-in" concept is presented as a classification problem. Experimental results show the potential of six different input
cases. When combining more 3D map information, the accuracy increases. To overcome the lack of training data, we produced the dataset by projecting from the 3D environment model lower the collecting cost dramatically. Benefit from B-CNN architecture, the coarse classification assists the model to accomplish fine classification tasks and achieve higher accuracy. For future work, more advanced denoise can be applied to the training data. By modeling the effects of noise, the proposed approach can be a sound choice for the visual indoor positioning system.

REFERENCES


