Scene Visual Perception and AR Navigation Applications

LU Ping[1,2], SHENG Bin[2], SHI Wenzhe[1,2]
(1. State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen 518055, China;
2. ZTE Corporation, Shenzhen 518057, China;
3. Shanghai Jiao Tong University, Shanghai 200240, China)

DOI: 10.12142/ZTECOM.202301010
https://kns.cnki.net/kcms/detail/34.1294.TN.20230215.1801.002.html, published online February 16, 2023
Manuscript received: 2022-11-01

Abstract: With the rapid popularization of mobile devices and the wide application of various sensors, scene perception methods applied to mobile devices occupy an important position in location-based services such as navigation and augmented reality (AR). The development of deep learning technologies has greatly improved the visual perception ability of machines to scenes. The basic framework of scene visual perception, related technologies and the specific process applied to AR navigation are introduced, and future technology development is proposed. An application (APP) is designed to improve the application effect of AR navigation. The APP includes three modules: navigation map generation, cloud navigation algorithm, and client design. The navigation map generation tool works offline. The cloud saves the navigation map and provides navigation algorithms for the terminal. The terminal realizes local real-time positioning and AR path rendering.

Keywords: 3D reconstruction; image matching; visual localization; AR navigation; deep learning

Citation (IEEE Format): P. Lu, B. Sheng, and W. Z. Shi, "Scene visual perception and AR navigation applications," *ZTE Communications*, vol. 21, no. 1, pp. 81 – 88, Mar. 2023. doi: 10.12142/ZTECOM.202301010.

1 Introduction

Navigation services applied to mobile devices are an indispensable part of modern society. At present, the outdoor positioning and navigating service technology has become mature, and the Global Positioning System (GPS) can provide relatively accurate position information and related supporting navigation services for outdoor pedestrians. For example, the navigation products of Baidu, Amap, Tencent and other companies can meet the location information and navigation service needs of outdoor pedestrians in terms of location services. However, once pedestrians go indoors, e.g., in shopping malls, airports, underground parking lots and other sheltered places, the positioning signal is greatly attenuated by factors like walls, and the GPS-based outdoor navigation technology becomes insufficient. The existing indoor localization methods have many constraints in localization accuracy, deployment overhead, and resource consumption, which limits their promotion in real-world navigation applications.

In recent years, researchers have designed a variety of indoor and outdoor positioning solutions for various types of information such as visible light communication (VLC), built-in sensors, QR codes, and WIFI. However, these solutions have many shortcomings in terms of localization accuracy, deployment difficulty, and equipment overhead. For example, the VLC-based methods require indoor LED lights to be upgraded on a large scale, which greatly increases deployment costs. Meanwhile, the WIFI-based methods cannot provide accurate direction information, which is difficult to meet the needs of precise localization.

However, in a visual scenario perception method, target recognition and position calculation are performed by means of image processing, so that relatively high positioning precision can be provided, and deployment of an additional device is not required, which is widely researched and applied in recent years.

The main application of scene perception is visual localization, which is a method of determining the position of 6-degree of freedom (6-DoF) from the image. The initialization conditions of visual localization usually require a sparse model of the scene and the estimated pose of the query image. Augmented reality (AR) navigation is an important application scenario of visual localization technologies, which can interact with the real world in a virtual environment through localization. The application of AR navigation technologies has great prospects in the future. Shopping malls have the most demand for localization and navigation technologies, and users are very interested in store discount information, personalized advertisements, store ratings, store locations, and indoor road

This work was supported by ZTE Industry-University-Institute Cooperation Funds under Grant No. HC-CN-20210707004.

guidance. The application of scene visual perception and AR navigation can solve most of the above problems well, and has vast potential in future development in the expansion of added value.

This paper introduces the design and implementation of AR navigation applications (APPs) and the cloud algorithm in detail, and starts from three aspects: navigation map generation, the cloud navigation algorithm, and the client design. Combined with specific cases, this paper introduces in detail the process of panoramic data acquisition and processing, point cloud map[1] and computer aided design (CAD) map alignment in the navigation map generation tool, and introduces the path planning algorithm and path correction algorithm in the cloud navigation algorithm. In terms of localization and AR path rendering, the client design method is introduced in detail, and finally, the running example of an AR navigation APP is given.

2 Basic Framework of Scene Visual Perception

Similar to humans, machines perceive and understand the environment mostly through visual information. In recent years, the development of 3D visual perception methods has provided great help for building models of the real physical world. For various application scenarios, there are currently some vision algorithms with commercial application capabilities, including face recognition, living body detection, 3D reconstruction, simultaneous localization and mapping (SLAM), gesture recognition, behavior analysis, augmented reality, virtual reality, etc.

Scene visual perception applied to navigation mainly includes 3D reconstruction and SLAM. The above steps can be regarded as the process of building a visual map. Visual map-based localization usually includes steps such as visual map construction and update, image retrieval, and fine localization, among which the visual map is the core of the method. According to the condition that the image frame has accurate prior pose information or not, the process of constructing a visual map can be divided into prior pose-based construction methods and non-prior pose methods. In the prior pose-based construction methods, the prior pose of the image frame can be derived from the high-precision LiDAR data synchronized and calibrated with the camera, which is common in high-precision acquisition vehicles in the field of autonomous driving. In small-scale scenes, especially indoors, the prior pose can also be obtained from visual motion capture systems such as Vicon and OptiTrack. The non-prior pose methods adopt offline extraction of feature points and offline optimization of pose and scene structures, which is similar to structure-from-motion (SfM). The constructed geometric visual map generally includes image frames, feature points and descriptors, 3D points, the correspondence between image frames, and the correspondence between 2D points and 3D points. During the process, due to changes in the real scene, the constructed visual map also needs to be updated synchronously to detect new and expired changes in time, and then update the corresponding changes to the visual map. When the prior visual map is obtained, the image retrieval and fine localization steps can usually be performed on the newly acquired image frame to complete localization. In the visual map-based localization framework, sensor information such as inertial measurement unit (IMU), GPS, and wheel odometer can also be fused.

3 Introduction to Key Technologies of Scene Visual Perception

3.1 3D Reconstruction

Accurate and robust 3D reconstruction methods are crucial to visual localization. The purpose of 3D reconstruction is to obtain the geometry and structure of an object or a scene from a set of images. SfM is a way to achieve 3D reconstruction, which is mainly used in the stage of building sparse point cloud in 3D reconstruction. A complete 3D reconstruction process usually also includes a multi-view stereo (MVS) step to achieve dense reconstruction. SfM is mainly used for mapping and restoring the structure of the scene. According to the difference in the image data processing flow, SfM can usually include four categories: incremental SfM, global SfM, distributed SfM, and hybrid SfM. Among them, distributed SfM and hybrid SfM are usually used to solve large-scale reconstruction and are based on incremental SfM and global SfM. Incremental SfM mainly includes two steps. The first step is to find the initial correspondence, and the second step is to achieve incremental reconstruction. The former aims to extract robust and well-distributed features to match image pairs, and the latter is used to estimate the image pose and 3D structure through image registration, triangulation, bundle adjustment (BA), and outlier removal. The initial corresponding outliers usually need to be removed by geometric verification methods. Generally, when the number of recovered image frames accounts for a certain proportion, global BA is required. Due to the incremental BA processing, incremental SfM usually has higher accuracy and better robustness. As the number of images increases, the scale of BA processing becomes larger, leading to disadvantages such as low efficiency and large memory usage. Additionally, incremental SfM suffers from cumulative drift as images are incrementally added. Typical SfM frameworks include Bundler and COLMAP.

CAO et al.[2] proposed a fast and robust feature tracking method for 3D reconstruction using SfM. First, to save computational costs, a feature clustering method was used to cluster a large set of images into small ones to avoid some wrong feature matching. Second, the joint search set method was used to achieve fast feature matching, which could further save the computational time of feature tracking. Third, a geometric constraint method was proposed to remove outliers in trajectories produced by feature tracking methods. The method could cope with the effects of image distortion, scale changes, and illumi-

nation changes. LINDENBERGER et al.[3] directly aligned low-level image information from multiple views, optimized feature point locations using depth feature metrics after feature matching, and performed BA through similar depth feature metrics during incremental reconstruction. In this process, the convolutional network was used to extract the dense feature map from the image, then the position of the feature points in the image was adjusted according to the sparse feature matching to obtain the two-dimensional observation of the same 3D point in different images, and the SfM reconstruction was completed according to the adjustment. The BA optimization residual in the reconstruction process changes from reprojection error to feature metric error. This improvement is robust to large detection noise and appearance changes, as it optimizes feature metric errors based on dense features predicted by neural networks.

The cumulative drift problem can be solved by global SfM. For the fundamental and essential matrix between images obtained in the image matching process, the relative rotation and relative translation can be obtained through decomposition. Using the relative rotation as a constraint, the global rotation can be recovered, and then the global translation can be recovered using the global rotation and relative translation constraints. Since the construction of the global BA does not require multiple optimizations, the global SfM is more efficient. However, since the relative translation constraints only constrain the translation direction and the scale is unknown, the translation averaging is difficult to solve. In addition, the translational average solution process is sensitive to outliers, so the global SfM is limited in practical applications.

3.2 Image Matching

How to extract robust, accurate, and sufficient image correspondences is a key issue in 3D reconstruction. With the development of deep learning, learning-based image matching methods have achieved excellent performance. A typical image matching process usually includes three steps: feature extraction, feature description, and feature matching.

Detection methods based on deep convolutional networks search for interest points by constructing response graphs, including supervised methods[4-5], self-supervised methods[6-7], and unsupervised methods[8-9]. Supervised methods use anchors to guide the training process of the model, but the performance of the model is likely to be limited by the anchor construction method. Self-supervised and unsupervised methods do not require human-annotated data, while they focus on geometric constraints between image pairs. Feature descriptors use local information around interest points to establish the correct correspondence of image features. Due to the information extraction and representation capabilities, deep learning techniques have also achieved good performance in feature descriptions. The deep learning-based feature description problem is usually a supervised learning problem, that is, learning a representation so that the matched features in the measurement space are as close as possible, and the unmatched features are as far as possible[10]. Learning-based descriptors largely avoid the requirement of human experience and prior knowledge. Existing learning-based feature description methods include two categories, namely metric learning[11-12] and descriptor learning[13-14], and the difference lies in the output content of the descriptor. Metric learning methods learn metric discriminants for similarity measurement, while descriptor learning generates descriptor representations from raw images or image patches.

Among these methods, SuperGlue[14] proposed a network capable of feature matching and filtering outliers simultaneously, whose feature matching was achieved by solving a differentiable optimization transfer problem. The loss function was constructed by a graph neural network, and a flexible content aggregation mechanism was proposed based on the attention mechanism, which enabled SuperGlue to simultaneously perceive potential 3D scenes and perform feature matching. LoFTR[15] used a transformer module with self-attention and cross-attention layers to process dense local features extracted from convolutional networks. Dense matches were first extracted at a low feature resolution (1/8 of the image dimension), from which high-confidence matches were selected and refined to high-resolution sub-pixel levels using correlation-based methods. In this way, the large receptive field of the model enabled the transformed features to reflect context and location information, and the prior matching was achieved through multiple self-attention and cross-attention layers. Many methods integrate feature detection, feature description, and feature matching into matching pipelines in an end-to-end manner, which is beneficial for improving matching performance.

3.3 Visual Localization

Visual localization is a problem of estimating the pose of a 6-DoF camera, from which a given image is obtained relative to a reference scene representation. Classical approaches to visual localization are structure-based, which means that they rely on 3D reconstructions of the environment (e. g. point clouds) and use local feature matching to establish correspondences between query images and 3D maps. Image retrieval can be used to reduce the search space by considering only the most similar reference images instead of all possibilities. Another approach is to directly interpolate the pose from the reference image or estimate the relative pose between the query and the retrieved reference image, which does not rely on the 3D reconstruction results. Scene point regression methods can directly obtain the correspondence between 2D pixel positions and 3D points using a deep neural network (DNN), and compute camera poses similar to structure-based methods. Modern scene point regression methods benefit from 3D reconstruction during training but do not rely on it. Absolute pose regression methods use a DNN to estimate poses end-to-

end. These methods differ in generalization ability and localization accuracy. Furthermore, some methods rely on 3D reconstruction, while others only require pose-labeled reference images. The advantage of using 3D reconstructions is that the generated poses can be very accurate, while the disadvantage is that these 3D reconstructions are sometimes difficult to obtain and even more difficult to maintain. For example, if the environment changes, they need to be updated.

The typical work of the structure-based approach can refer to a general visual localization pipeline proposed in Ref. [17]. Through a hierarchical localization approach, the pipeline can simultaneously predict local features and global descriptors for accurate 6-DoF localization, which utilizes a coarse-to-fine localization paradigm, first performing global retrieval to obtain location hypotheses and then matching local features in these candidate locations. This hierarchical approach saves runtime for real-time operations and proposes a hierarchical feature network (HF-Net) that jointly estimates local and global features, thereby maximizing shared computation, and compresses the model through multi-task distillation.

4 AR Navigation Based on Scene Visual Perception

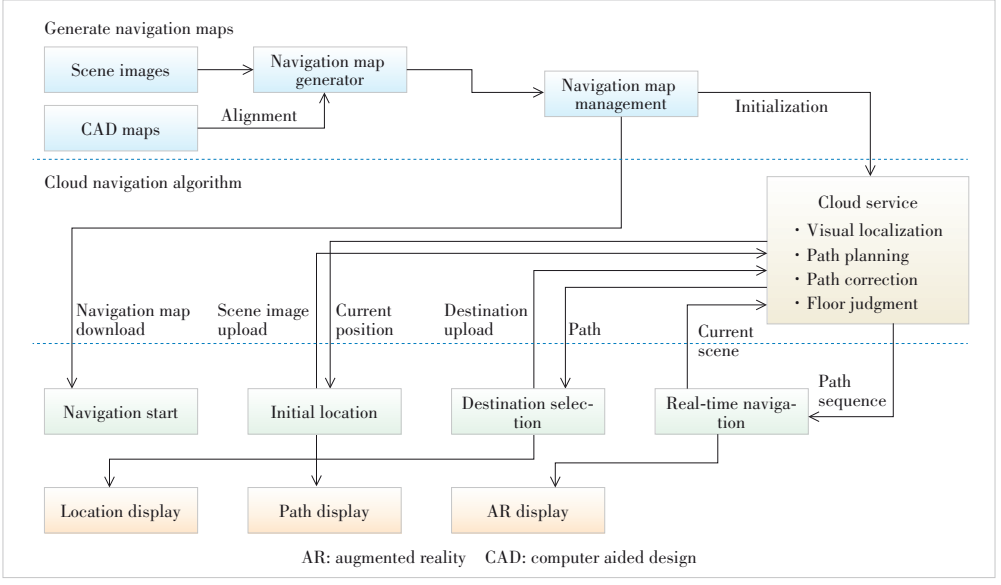
AR navigation usually works in the following process: 1) The real-world view is got from the user's point of view; 2) the location information is obtained and used to track the user; 3) virtual-world information is generated based on the real-world view and location information; 4) the generated virtual world information is registered into the real-world view and displayed to the user, creating augmented reality. The main challenge of AR navigation is how to integrate the virtual and real worlds, and design and present the navigation interface. Registration is the process of correctly aligning virtual information with the real world, which gives the user the illusion of keeping the virtual and the real coexisting. For AR in navigation, accurate registration is critical, and AR navigation systems can cause confusion when orientation changes rapidly due to registration errors. So even small offsets of registering dummy information can be harmful. In an AR navigation system, the display should not interfere with the user's movement. The augmented reality display technology is also known as video see-through. Video see-through display refers to placing a digital screen between the real world and the user, where the user can see the real world and augmented information, use a camera to capture the real-world view, and then combine it with the augmented information and display it on the screen superior. Typical examples of displays include head-mounted displays with cameras and smartphone displays.

On the basis of scene visual perception, this paper designs an AR navigation APP developed based on Unity and ARCore. Its overall framework is shown in Fig. 1. The system consists of three parts, namely, the navigation map generation tool, the cloud navigation algorithm, and the terminal navigation APP design.

The navigation map generation tool works offline, including scene panoramic video capture, dense point cloud generation, point cloud and plane CAD map alignment, navigation map management and other functions. The map generated by the navigation map generation tool is stored in the cloud. In addition, the cloud is also responsible for providing navigation algorithms to the terminal, including visual localization methods, path planning algorithms, path correction algorithms, floor judgment algorithms and cross-layer guidance algorithms. When users request a navigation activity with the terminal APP, they first select the current location map, and the cloud issues the corresponding navigation map according to the user's selection. After selecting the starting point and ending point, the user requests the navigation service from the cloud, and realizes local real-time localization, global path and current position display, and AR path rendering in the local APP.

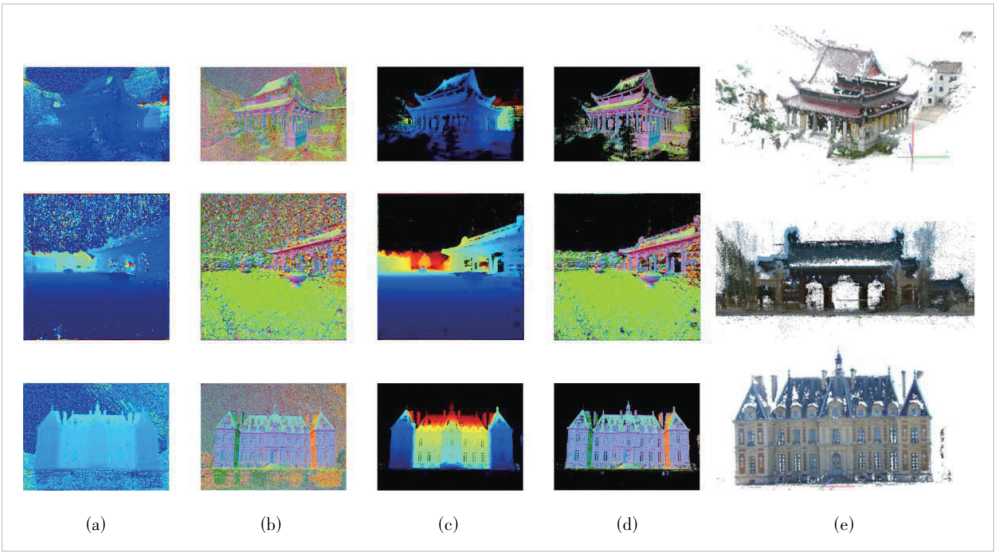
4.1 Panoramic Data Collection and Processing

This paper uses a panoramic camera to capture video to collect mapping data. Instead of rotating the camera around its optical center, this panoramic camera can be used to capture



▲Figure 1. Overall framework of an AR navigation application (APP)

multiple images of a scene from different viewpoints, from which stereoscopic information about the scene can be calculated. The stereo information is then be used to create a 3D model of the scene, and arbitrary views can be computed. This approach is beneficial for 3D reconstruction of large-scale scenes. The dense reconstruction results of the proposed approach on the building dataset are shown in Fig. 2.



▲**Figure 2. Result of dense reconstruction: (a) photometric depth map, (b) photometric normal map, (c) geometric depth map, (d) geometric normal map, and (e) dense reconstruction effect**

Taking a large shopping mall as an example, for the processing and 3D reconstruction of the data collected from the panoramic video, this paper goes through the following steps:

1) Shoot a panoramic video of the scene, and the shooting area should be covered as much as possible;

2) Frame the obtained panoramic video to obtain a panoramic image and segment the panoramic image according to the field of view (FOV);

3) Realize sparse point cloud reconstruction for each floor and finally output all camera parameters and sparse 3D point cloud;

4) Complete the single-layer dense point cloud reconstruction;

5) Integrate multiple layers of dense point clouds to obtain a complete 3D structure of the scene.

4.2 Alignment of Point Cloud Map and CAD

The point cloud obtained in Section 3.1 is based on the camera coordinate system, which must be aligned with the world coordinate system if it is to be used for navigation tasks. This paper takes the CAD map as the world coordinate system, because CAD can provide accurate position information and scale information. The problem is transformed into the alignment of the point cloud map and the plane CAD. The specific process of its realization is as follows:

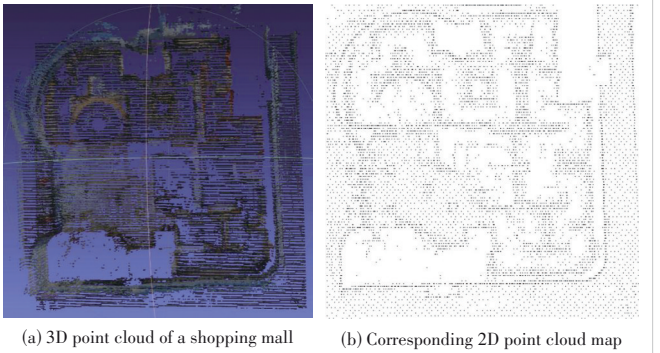
1) The point cloud is dimensionally reduced and projected to the XoY plane to form a plane point cloud map, as shown in Fig. 3.

2) Marker points (such as walls and other points that are easy to be distinguished) and the corresponding points are found on the plane point cloud map and the CAD map, respectively.

3) Alignment is completed through the scale information provided by the CAD map, output rotation and the displacement matrix.

Once the point cloud $\boldsymbol{X}$ is sampled, it can be mapped to a 2D plane by simply removing the z coordinates. The problem is transformed into finding the mapping between $(\boldsymbol{X}_x, \boldsymbol{X}_y)$ and pixels (u, v), where $(\boldsymbol{X}_x, \boldsymbol{X}_y)$ is the set of 2D coordinates (x, y) extracted from the point cloud X. It is worth noting that (x, y) are usually float values, while pixel coordinates (u, v) are usually positive integer values. Therefore, (x, y) needs to go through a certain scale, rotation and rounding transformation.

Once the plane point cloud map is obtained, it can be aligned with the CAD map through the affine transformation. To determine the affine matrix, at least three pairs of corresponding points are usually required. Considering the need to reduce errors, this paper selects multiple pairs of corresponding points in the point cloud map and CAD map respectively, and uses the least square method to achieve alignment. It is worth noting that the selection of corresponding points should try to select parts that are easy to identify, such as walls and other fixed objects with clear structural characteristics. Fig. 3 shows the process of aligning a point cloud map with a CAD map. After the alignment, the position coordinates of the point cloud in the world coordinate system can be obtained, which



▲**Figure 3. An example of a 2D point cloud map generation**

is beneficial to the subsequent localization and navigation tasks. The obtained results can be saved separately according to the scene, and the saved content includes the scene pose, corresponding geographic information, camera model, and other information to form a navigation digital map.

4.3 Cloud Navigation Algorithm

When a user requests a navigation activity with the terminal APP, he first selects the map corresponding to the current location, and the cloud issues the corresponding navigation map according to the user's selection. After the user selects the destination, the user requests the navigation service from the cloud, and at the same time uploads the current scene graph to the cloud. At this time, the cloud needs to invoke the visual localization algorithm to determine the current initial position of the user as a starting point. After obtaining the coordinates of the starting point and the ending point, the cloud calls the path planning algorithm to obtain the navigation path point sequence and sends it to the terminal APP for AR rendering. The user is actually positioned through ARCore during the process of traveling. However, this method will generate accumulated errors after traveling for a certain distance, and since the user may deviate from the recommended path, the path correction algorithm needs to be implemented through the cloud, and the user is directed to the correct path.

According to common practice in the industry, the path planning algorithm designed in this paper does not need to provide a path from any point to any point. The path planning involved in this paper only needs to provide a path from any point (user location or user-selected location) to a specific point (specified end-point set). Therefore, the path planning problem in this paper can be regarded as solving the shortest path problem between the vertices of a directed graph. The basic flow of the path planning algorithm proposed in this paper is as follows:

1) The passable area is determined through the point cloud map, and the waypoint is selected in the passable area.

2) The route point and the destination point (the selected end-point) form a graph structure.

3) The shortest path is found among all vertices in the graph through a search algorithm.

The process of building route points and destination points into a graph structure forms a road network. In this process, it is necessary to clarify the world coordinates of the waypoint and the destination point, and mark the connection relationship between points to form a graph structure of the road network, which is stored in the form of an adjacency list. Since the purpose of this paper is to find the shortest path among all vertices in the graph, it constitutes an all pairs shortest paths (APSP) problem. The general solution to the APSP problem is the Floyd-Warshall algorithm. After the shortest path among all points is obtained, the result is saved in the cloud according to the scene, so that in practical applications, there is no need to calculate the planned path online, and only the retrieval function will be implemented, which is time-consuming.

During the user's journey, the local positioning provided by ARCore will gradually produce errors with the advancing distance. At the same time, the user may deviate from the recommended navigation path due to internal or external reasons. Therefore, the cloud needs to provide a path correction algorithm to guide the user back to the navigation path (the correct path). The specific workflow of the path correction algorithm is as follows:

1) The user uploads the current scene image while traveling.

2) The cloud determines whether it deviates from the navigation path recommended by the algorithm according to the positioning algorithm.

3) If the user's deviation is small, the user will be guided to the recommended navigation path through the navigation arrows of the terminal APP. If the user's deviation is too large, the path planning will be re-planned based on the user's current position.

The path correction process is actually a verification process of the real-time local positioning information fed back by the terminal. When the error exceeds the distance threshold τ, the path correction function can be activated. In practical applications, the selection of the distance threshold τ is usually between 50 cm and 200 cm. If the threshold is too small, it will increase the influence of visual positioning errors. If the threshold is too large, it will not only lose the accuracy of navigation, but also bring inconvenience to users.

4.4 AR Systems

AR systems contain three basic features: the combination of real and virtual worlds, real-time interaction, and accurate 3D registration of virtual and real objects. In this way, AR changes people's continuous perception of the real environment and obtains an immersive experience by integrating the composition of the virtual world into people's perception of the real environment. Specific to AR navigation APPs, users can obtain real-world information from smartphones (through the phone camera), and by applying the AR technology, virtual navigation paths can be added to the smartphone's interface, enhancing the user's perception of the real environment for a better navigation experience. From the user's point of view, a complete AR navigation includes the following process: 1) The user selects the current scene and obtains the navigation map delivered by the cloud; 2) the user selects the destination according to the navigation map and requests the cloud navigation service; 3) the user follows the terminal interface rendering AR path to the end. Due to network bandwidth limitations, users cannot obtain real-time localization by sending the current scene image to the cloud in real time. Therefore, the ARCore-based method is used to provide real-time localization. However, this method will generate accumulated er-

rors after traveling for a certain distance. And since users may deviate from the recommended path, path correction needs to be implemented through a correction algorithm to guide users to the correct path. Fig. 4 shows the flow of the AR navigation APP and AR rendering.

ARCore is an AR application platform provided by Google, which can be easily combined with 3D engines such as Unreal and Unity. ARCore provides three main applications for motion tracking, environment understanding, and lighting estimation. Among them, motion tracking enables the phone to know and track its position relative to the world, environment understanding enables the phone to perceive the environment, such as the size and location of detectable surfaces, and light estimation allows the phone to obtain the current lighting conditions of the environment. Localization can be achieved using ARCore's motion-tracking capabilities.

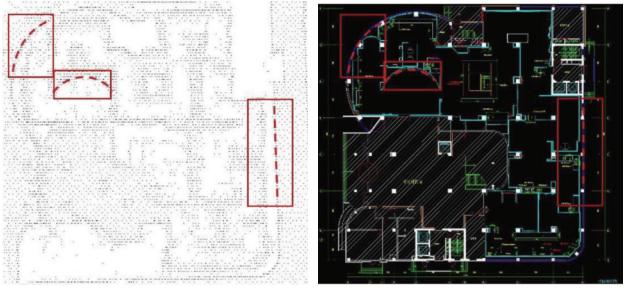
The motion-tracking function of ARCore is actually realized by visual inertial odometry (VIO). VIO includes two parts: a visual tracking system and an inertial navigation system. The camera obtains a frame of pixel matching to track the user's pose. The inertial navigation system realizes position and attitude tracking through an IMU, which usually consists of an accelerometer and a gyroscope. The outputs of the two systems are combined through a Kalman filter to determine the final pose of the user. The local positioning function provided by ARCore can track the user's position in real time, but the error in the inertial navigation system of ARCore will accumulate over time. As the user's advancing distance increases and time passes, tracking of the user's position will be offset. In practice, we find that after a user travels about 50 m, the localization provided by ARCore will begin to deviate. At this time, it is necessary to relocate through the visual localization algorithm and correct the path.

On the basis of the previous work, the AR navigation APP can obtain the current position of the user and the path point sequence of the path planning from the cloud. Then the next question is how to realize AR rendering of the path point sequence on the mobile phone interface. From the perspective of user experience, the AR markers cannot block the user's line of sight and must provide an obvious guiding role. Therefore, in the actual rendering process, this paper chooses to render the AR markers close to the ground. The environment understanding section in ARCore provides plane detection capabilities. In fact, ARCore stipulates that all virtual objects need to rely on planes for rendering. After ARCore implements plane detection, the AR markers can be placed on the ground. The placement of AR markers can be achieved by radiographic inspection. The principle of ray detection is to judge whether there is a collision with an object through the ray emitted from the camera position to any position in the 3D world. In this way, the collision object and its position can be detected. By performing collision detection on the planes in the scene, the planes can be judged and AR signs can be placed. Here, this paper adopts two kinds of AR markers, one is the navigation guidance arrow, which is responsible for indicating the forward direction, and the other is the end prompt sign, which reminds the user to reach the end-point. Fig. 4 shows the actual workflow of the AR navigation APP and the rendering effect of the AR markers. In the figure, from left to right, the user selects the destination (elevator entrance), the navigation guide arrow is rendered, the user follows the navigation guide arrow, and the navigation ends at the end prompt sign.

5 Conclusions and Outlook

This paper analyzes and introduces related technologies in the field of scene visual perception, based on which we implement AR navigation. In practical application, there are still some problems to be solved[18-19]. For example, this paper adopts a structure-based localization framework, with an advantage that it can effectively handle large-scale scenes and has high localization accuracy. However, if the environment changes, the 3D structure needs to be re-adjusted to achieve re-registration of point clouds. The alignment method of point cloud map and plane CAD shown in Fig. 5 still requires manual selection of corresponding points, which is not conducive to large-scale applications, so it needs to be studied in

▲Figure 4. Augmented reality (AR) navigation application (APP) and AR rendering result



(a) Corresponding points in point cloud map

(b) Corresponding points in CAD map

CAD: computer aided design

▲Figure 5. An example of a 2D point cloud map aligned with CAD map

the follow-up work to realize the automatic process. The proposed localization method in this paper adopts a pure vision solution. In the future, it can also be considered to combine other sensor data such as IMU, depth camera or LiDAR to further improve the localization and navigation performance. In addition, most of the current visual localization algorithms cannot be independent of the scene, and usually need to train different models on different datasets (such as training models on indoor and outdoor datasets), which brings difficulties to practical applications. For example, in the AR navigation process, image feature matching is usually performed in the cloud. Due to the diversity of the user's scene, if a scene-related localization algorithm is used, the generalization ability of the model will be insufficient, which will lead to poor localization performance. Therefore, for AR navigation, it is particularly important to enhance the generalization performance of localization algorithms and achieve scene-independent visual localization.

References

[1] LI H Q, LI L, LI Z. A review of point cloud compression [J]. ZTE technology journal, 2021, 27(1): 5 – 9. DOI: 10.12142/ZTETJ.202101003
[2] CAO M, WEI J, LYU Z, et al. Fast and robust feature tracking for 3D reconstruction [J]. Optics & laser technology, 2019, 110: 120 – 128. DOI: 10.1016/j.optlastec.2018.05.036
[3] LINDENBERGER P, SARLIN P E, LARSSON V, et al. Pixel-perfect structure-from-motion with featuremetric refinement [C]//Proceedings of 2021 IEEE/CVF International Conference on Computer Vision. IEEE, 2022: 5967 – 5977. DOI: 10.1109/ICCV48922.2021.00593
[4] YI K M, TRULLS E, LEPETIT V, et al. LIFT: learned invariant feature transform [C]//European Conference on Computer Vision. ECCV, 2016: 467 – 483. DOI: 10.1007/978-3-319-46466-4_28
[5] ZHANG X, YU F X, KARAMAN S, et al. Learning discriminative and transformation covariant local feature detectors [C]//Conference on Computer Vision and Pattern Recognition. IEEE, 2017: 4923 – 4931. DOI: 10.1109/CVPR.2017.523
[6] ZHANG L G, RUSINKIEWICZ S. Learning to detect features in texture images [C]//Conference on Computer Vision and Pattern Recognition. IEEE, 2018: 6325 – 6333. DOI: 10.1109/CVPR.2018.00662
[7] DETONE D, MALISIEWICZ T, RABINOVICH A. SuperPoint: self-supervised interest point detection and description [C]//Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. IEEE, 2018: 224 – 236. DOI: 10.1109/CVPRW.2018.00060
[8] LAGUNA A B, RIBA E, PONSA D, et al. Key.Net: keypoint detection by handcrafted and learned CNN filters [C]//International Conference on Computer Vision (ICCV). IEEE, 2020: 5835 – 5843. DOI: 10.1109/ICCV.2019.00593
[9] ONO Y, TRULLS E, FUA P, et al. LF-Net: Learning local features from images [C]//Proceedings of the 32nd International Conference on Neural Information Processing Systems. ACM, 2018: 6237 – 6247. DOI: 10.5555/3327345.3327521
[10] SCHÖNBERGER J L, HARDMEIER H, SATTLER T, et al. Comparative evaluation of hand-crafted and learned local features [C]//Conference on Computer Vision and Pattern Recognition. IEEE, 2017: 6959 – 6968. DOI: 10.1109/CVPR.2017.736
[11] WANG J, ZHOU F, WEN S L, et al. Deep metric learning with angular loss [C]//International Conference on Computer Vision. IEEE, 2017: 2612 – 2620. DOI: 10.1109/ICCV.2017.283
[12] ZAGORUYKO S, KOMODAKIS N. Learning to compare image patches via convolutional neural networks [C]//Conference on Computer Vision and Pattern Recognition. IEEE, 2015: 4353 – 4361. DOI: 10.1109/CVPR.2015.7299064
[13] LUO Z X, SHEN T W, ZHOU L, et al. ContextDesc: local descriptor augmentation with cross-modality context [C]//Conference on Computer Vision and Pattern Recognition. IEEE, 2020: 2522 – 2531. DOI: 10.1109/CVPR.2019.00263
[14] TIAN Y R, YU X, FAN B, et al. SOSNet: second order similarity regularization for local descriptor learning [C]//Conference on Computer Vision and Pattern Recognition. IEEE, 2020: 11008 – 11017. DOI: 10.1109/CVPR.2019.01127
[15] SARLIN P E, DETONE D, MALISIEWICZ T, et al. SuperGlue: learning feature matching with graph neural networks [C]//Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2020: 4937 – 4946. DOI: 10.1109/CVPR42600.2020.00499
[16] SUN J M, SHEN Z H, WANG Y A, et al. LoFTR: detector-free local feature matching with transformers [C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2021: 8918 – 8927. DOI: 10.1109/CVPR46437.2021.00881
[17] SARLIN P E, CADENA C, SIEGWART R, et al. From coarse to fine: robust hierarchical localization at large scale [C]//Conference on Computer Vision and Pattern Recognition. IEEE, 2020: 12708 – 12717. DOI: 10.1109/CVPR.2019.01300
[18] LU P, SHENG B, ZHU F. Next-generation communications technology facilitates real-time distributed cloud rendering [J]. ZTE technology journal, 2021, 27(1): 17 – 20. DOI: 10.12142/ZTETJ.202101005
[19] LU P, OUYANG X Z, GAO W W. Capacity improvement and practice of 5G industry virtual private network [J]. ZTE technology journal, 2022, 28(2): 68 – 74. DOI: 10.12142/ZTETJ.202202011

Biographies

LU Ping is the Vice President and general manager of the Industrial Digitalization Solution Department of ZTE Corporation, and Executive Deputy Director of the National Key Laboratory of Mobile Network and Mobile Multimedia Technology. His research directions include cloud computing, big data, augmented reality, and multimedia service-based technologies. He has supported and participated in major national science and technology projects and national science and technology support projects. He has published multiple papers, and authored two books.

SHENG Bin (shengbin@cs.sjtu.edu.cn) is a professor of computer science and engineering from Shanghai Jiao Tong University, China. His research directions include virtual reality and computer graphics. He has presided over two projects on the National Natural Science Foundation of China, one youth project of the National Natural Science Foundation of China, and participates in one high-technology research and development plan (the "863" plan) and one key project of the National Natural Science Foundation of China. He has published 121 papers in different journals.

SHI Wenzhe is a strategy planning engineer with ZTE Corporation, a member of the National Key Laboratory for Mobile Network and Mobile Multimedia Technology, and an engineer of XRExplore Platform Product Planning. His research interests include indoor visual AR navigation, SFM 3D reconstruction, visual SLAM, real-time cloud rendering, VR, and spatial perception.