Local Scenario Perception and Web AR Navigation

SHI Wenzhe[1,2], LIU Yanbin[1,2], ZHOU Qinfen[1]

(1. State Key Laboratory of Mobile Network and Mobile Multimedia Technology, Shenzhen 518055, China;
2. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202304007

https ://kns.cnki.net/kcms/detail/34.1294.TN.20231129.1522.004.html, published online November 30, 2023

Manuscript received: 2023-06-20

Abstract: This paper proposes a local point cloud map-based Web augmented reality (AR) indoor navigation system solution. By delivering the local point cloud map to the web front end for positioning, the real-time positioning can be implemented only with the help of the computing power of the web front end. In addition, with the characteristics of short time consumption and accurate positioning, an optimization solution to the local point cloud map is proposed, which includes specific measures such as descriptor de-duplicating and outlier removal, thus improving the quality of the point cloud. In this document, interpolation and smoothing effects are introduced for local map positioning, enhancing the anchoring effect and improving the smoothness and appearance of user experience. In small-scale indoor scenarios, the positioning frequency on an iPhone 13 can reach 30 fps, and the positioning precision is within 50 cm. Compared with an existing mainstream visual-based positioning manner for AR navigation, this specification does not rely on any additional sensor or cloud computing device, thereby greatly saving computing resources. It takes a very short time to meet the real-time requirements and provide users with a smooth positioning effect.

Keywords: Web AR; three-dimensional reconstruction; navigation; positioning

Citation (Format 1): SHI W Z, LIU Y B, ZHOU Q F. Local scenario perception and Web AR navigation [J]. ZTE Communications, 2023, 21(4): 54 – 59. DOI: 10.12142/ZTECOM.202304007

Citation (Format 2): W. Z. Shi, Y. B. Liu, and Q. F. Zhou, "Local scenario perception and Web AR navigation," *ZTE Communications*, vol. 21, no. 4, pp. 54 – 59, Dec. 2023. doi: 10.12142/ZTECOM.202304007.

1 Introduction

There are three existing positioning manners for augmented reality (AR) navigation: visual positioning, GPS positioning, and multi-sensor fusion positioning. Specifically, visual positioning is based on a conventional simultaneous localization and mapping (SLAM) framework, and the main steps include feature extraction, feature matching, and position solving. This model has high precision, but requires high computing power and cannot run on the Web side.

Outdoor AR navigation can achieve good results by using GPS technology. However, in an indoor scenario, the signal strength of GPS is greatly affected, and consequently, positioning precision is obviously reduced. Therefore, the GPS-based positioning manner cannot be applied to indoor AR navigation.

A positioning manner of multi-sensor fusion is to obtain the position of a camera by fusing data from sensors such as an inertial sensor, a laser radar, Bluetooth, and Wi-Fi. In this manner, although positioning accuracy is high, the sensor is vulnerable to environments, thereby decreasing positioning performance. In addition, in this manner, a large number of sensors need to be calibrated and fused, and a development cost is high.

Although the foregoing three methods can obtain relatively high precision in certain specific scenarios, they cannot be applied to an indoor Web AR navigation scenario. The core reason is that the computing power on the Web side is limited and cannot meet the intensive requirements of AR computing. If a positioning system that meets performance requirements can be implemented on the web front end by using limited computing power, dependence on an external computing environment or device will be reduced, development costs can be cut, and the application scope and user experience of indoor Web AR navigation will be greatly improved.

2 Key Technologies of Visual Perception

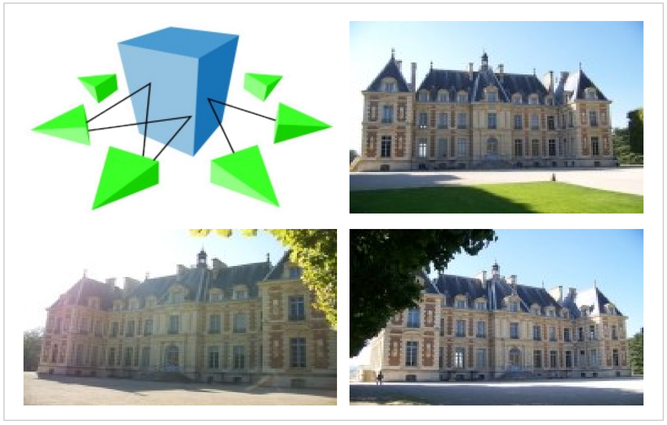
2.1 3D Reconstruction Techniques

To implement a visual method of good positioning, precision is indispensable for a robust three-dimensional reconstruction process. An objective of three-dimensional reconstruction is to obtain a geometric structure and a structure of an object or a scene from a group of images, which may be implemented by using a motion recovery structure (Structure-from-Motion, SFM). SFM is a method for implementing three-dimensional reconstruction and mainly used in a phase of con-

This work is supported in part by ZTE Industry-University-Institute Cooperation Funds.

structing a sparse point cloud image in the three-dimensional reconstruction. A complete three-dimensional reconstruction process generally uses a Multi-View Stereo (MVS) algorithm to implement dense reconstruction. As shown in Fig. 1, SFM is mainly used for creating diagrams and restoring the structure of the scenario. According to the difference of image data processing flows, SFM can be divided into four types: incremental SFM, global SFM, distributed SFM, and hybrid SFM. The latter two types are usually used to resolve a very large-scale data scenario and are based on the former two types. Incremental SFM can be divided into two steps. The first step is to find the initial correspondence to extract robust and well-distributed features to match the image pairs, and the second step is to implement incremental reconstruction to estimate image position and 3D structure by image registration, triangulation, bundle adjustment (BA), and abnormal value removal. The initial corresponding abnormal value needs to be removed through geometric verification. Generally, when the number of restored image frames accounts for a certain proportion, global BA is performed. Because of the incremental processing of BAs, the precision of the incremental SFM is usually relatively high and the robustness is relatively good. However, with the increase of the images, the processing scale of the BAs becomes larger and larger. Therefore, there are also disadvantages such as low efficiency and large memory usage. In addition, the incremental SFM also has the problem of accumulative drift because of the incremental addition of images. Typical SFM frameworks include Bundler and COLMAP.

CAO et al. [1] proposed a fast and robust feature-tracking method for 3D reconstruction using SFM. First, to reduce calculation costs, a large number of image sets are clustered into some small image sets by using a feature clustering method to avoid some incorrect feature matching. Second, a joint search set method is used to implement fast feature matching, which may further save calculation time of feature tracking. Third, a geometric constraint method is proposed to remove an abnormal value from a track generated by a feature tracking method. This method can deal with the influence of image distortion, scale change and illumination change. LINDENBERGER et al.[2] directly align low-level image information from multiple views, optimize feature point positions using depth feature metrics after feature matching, and perform BA during incremental reconstruction using similar depth feature metrics. In this process, an image-dense feature map is first extracted by using a convolution network, two-dimensional observation of the same three-dimensional point in different images is obtained using sparse feature matching, the location of a corresponding feature point in the image is adjusted, SFM reconstruction is performed according to the adjusted location, and a residual of SFM optimization in the reconstruction process changes from a reprojection error to a depth feature measurement error. This improvement is robust to large-scale detection of noise and appearance changes because it optimizes feature measurement errors for dense features based on neural network prediction.



▲Figure 1. Shooting a panoramic video of the scene

Some accumulated drift problems are solved through global SFM. In an image matching process, a basic/essential matrix between images is obtained, and relative rotation and relative translation between the images may be obtained by means of decomposition. Global rotation can be restored by using relative rotation as a constraint. Global panning can then be restored using the global rotation and relative panning constraints. Because the number of times of building and optimizing global BA is small, the efficiency of global SFM is high. However, it is difficult to solve the translation average because the relative translation constraint only constrains the translation direction and the scale is unknown. In addition, the translation average solving process is sensitive to external points. Therefore, in actual applications, the global SFM is limited.

2.2 Space Visual Matching Technology

How to extract robust, accurate and sufficient image correspondence is the key problem of 3D reconstruction. With the development of deep learning, the image matching methods based on learning achieve excellent performance. A typical image matching process is divided into three steps: feature extraction, feature description, and feature matching.

Detection methods based on deep convolution networks search for points of interest by building response maps, including the supervisory method[3 – 4], self-supervised method[5 – 6], and unsupervised method[7 – 8]. The supervisory approach uses an anchor to guide the training process of a model, but the performance of the model is likely limited by the anchor construction approach. Self-supervised and unsupervised methods do not require manual annotation of data, and they focus on geometric constraints between image pairs.

The feature descriptor uses local information around the point of interest to establish a correct correspondence between image features. Due to the ability of information extraction and representation, depth techniques have also performed well in the description of features. The feature description problem based on deep learning is usually a supervised learning problem, that is, learning a representation that makes matched features in the measurement space as close as possible and unmatched features as far as possible[9]. Learning-based descriptors largely avoid the need for human experience and prior knowledge. An existing feature description method based on learning is classified into two types: measurement learning[10 – 11] and descriptor learning[12 – 13]. A difference lies in the output content of a descriptor.

Metric learning methodology is used for metric discrimination for similarity measurement, while descriptor learning generates descriptor representations from the original image or image block. In these methods, SuperGlue[14] is a network that can perform feature matching and filter out extrinsic points at the same time, where feature matching is implemented by solving a differential optimization transfer problem, a loss function is constructed by using a graph neural network, and a flexible content aggregation mechanism is proposed based on an attention mechanism. Therefore, SuperGlue can simultaneously sense a potential three-dimensional scene and perform feature matching. LoFTR[15] uses transformer modules with a self-attentive layer and a cross-attentive layer to process dense local features extracted from the convolutional network by first extracting dense matches at low feature resolution (1/8 of the image dimension) and then selecting the matches with high confidence from those matches using a relevant method to refine them to a high-resolution sub-pixel level. In this way, the large acceptance field of the model enables the converted signature to reflect the context and location information, and the matching is implemented through multiple layers of self-attention and cross-attention. Many methods integrate feature detection, feature description, and feature matching into the matching pipeline in an end-to-end manner, which helps improve matching performance.

Visual orientation is a problem of estimating a 6-DoF camera pose from which a given image is taken relative to a reference scene representation. The classical approach to visual positioning is structure-based, meaning that they rely on the 3D reconstruction of the environment (that is, point clouds) and use local feature matching to establish a mapping relationship between the query image and 3D map. Image retrieval can be used to reduce the search space by only considering the most similar reference images rather than all possibilities. Another approach is to interpolate or estimate the relative posture between the queried and retrieved reference images directly from the reference images, which is independent of the 3D reconstruction results. In the scene point regression method, a correspondence between a two-dimensional pixel position and a three-dimensional point may be directly determined by using a deep neural network (DNN), and the position of a camera is calculated similarly to a structure-based method. Modern scene regression benefits from 3D reconstruction during training but does not depend on it. Finally, the absolute posture regression method uses DNN end-to-end posture estimation. These approaches differ in generalization capabilities and location accuracy.

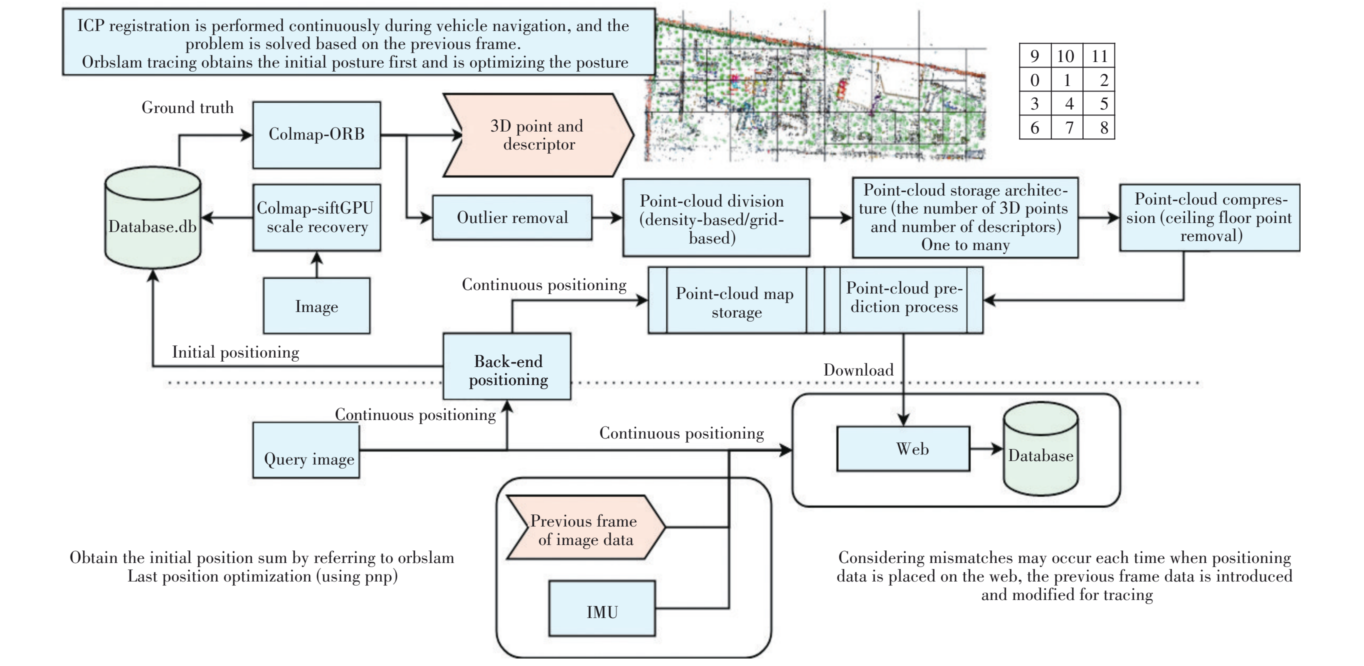
In addition, some methods rely on 3D reconstruction, while others only require reference images with position marks. The advantage of using a 3D reconstruction is that the position generated is very accurate, and the disadvantage is that these 3D reconstructions are sometimes difficult to obtain or even more difficult to maintain. For example, if the environment changes, the position needs to be updated. For classical structure-based work, reference may be made to a general visual positioning framework proposed by SARLIN et al.[16] The framework can predict both local features and global descriptors by using a hierarchical positioning method, so as to implement accurate 6-DoF positioning. Using a coarse-to-fine localization pattern, the method first performs a global search to obtain location assumptions and then matches local features in these candidate locations. This layered approach saves uptime for real-time operations. This method presents a hierarchical feature network (HF-Net), which jointly estimates local and global features, shares computation to the maximum extent, and uses a multitask still compression model.

3 Web AR Navigation System Based on Local Scenario Perception

This paper presents an indoor Web AR navigation system architecture based on the local point cloud map. By delivering the space local point cloud map to the web front end for positioning, the real-time positioning can be implemented only by using the computing power of the web front end, which has the characteristics of short time consumption and accurate positioning. In addition, this paper proposes an optimization solution to the local point cloud map, including specific measures such as descriptor deduplication and outlier elimination, which improves the quality of the point cloud. Finally, interpolation and smoothing effects are introduced to local map localization to enhance an anchoring effect and improve smoothness and appearance of user experience. In a small-scale indoor scenario, a localization frequency on an iPhone 13 may reach 30 fps, and localization precision is within 50 cm. In this paper, a function of implementing real-time positioning by using only Web front-end computing power is proposed for the first time. It outperforms existing mainstream visual-based positioning for AR navigation, GPS-based positioning, and multi-sensor fusion positioning. The proposed method can significantly save computing resources without the help of any additional sensors or cloud computing devices. It takes a very short time to meet the real-time requirements and provide users with smooth positioning, improving user experience.

Fig. 2 shows the proposed Web AR indoor navigation system based on local point cloud map positioning. This system consists of three modules: offline map creation, server, and web.

The offline map creation module is mainly responsible for the reconstruction of a point cloud map. Three-dimensional reconstruction is implemented by photographing an environmental image that needs to be reconstructed and then scale-based


ICP registration is performed continuously during vehicle navigation, and the problem is solved based on the previous frame.
Orbslam tracing obtains the initial posture first and is optimizing the posture
Ground truth
Colmap-ORB
3D point and descriptor
9 10 11
0 1 2
3 4 5
6 7 8
Database.db
Colmap-siftGPU scale recovery
Outlier removal
Point-cloud division (density-based/grid-based)
Point-cloud storage architecture (the number of 3D points and number of descriptors) One to many
Point-cloud compression (ceiling floor point removal)
Image
Continuous positioning
Point-cloud map storage
Point-cloud prediction process
Initial positioning
Back-end positioning
Download
Continuous positioning
Query image
Continuous positioning
Web
Database
Previous frame of image data
IMU
Obtain the initial position sum by referring to orbslam
Last position optimization (using pnp)
Considering mismatches may occur each time when positioning data is placed on the web, the previous frame data is introduced and modified for tracing

ICP: Iterative Closest Point IMU: Inertial Measurement Unit ORB: oriented FAST and rotated BRIEF

▲**Figure 2. Local scenario perception and the proposed web navigation system architecture**

restoration is performed, to finally obtain a sparse point cloud map and save the sparse point cloud map in the format of a 3D point plus a descriptor. Then, the point cloud is visualized and divided according to the preset interest point when the user wants to perform a model anchoring display. The related geo-fence information is set, which is mainly used for service experience after entering the local point cloud range. Currently, the geo-fence range is mainly 3 m – 5 m and established according to a specific scenario. The sparse point cloud is divided into multiple local point cloud maps. Next, the point cloud is optimized by using descriptor deduplication and outlier removal and is stored in the bin format.

The server performs positioning on the captured initial positioning picture to obtain an initial positioning posture of the camera, so as to determine a local point cloud closest to an initial positioning point position. The local point cloud communication service is responsible for delivering the specific local point cloud to the Web front end in accordance with the request of the Web front end for the local point cloud.

The Web front end sends a request to the server end by using a local point cloud communication service, receives specific local point cloud information, and then captures an image of a video frame by using a local point cloud positioning system of the Web front end. The Web front end obtains corresponding camera position information for positioning and then renders a navigation route and a corresponding material based on the camera position information obtained by positioning. Fig. 2 shows how to implement AR navigation through the local cloud.

The time consumption of each step of the local point cloud positioning algorithm is collected and optimized, including image data transmission on a Web end, improvement of a feature extraction algorithm, feature matching optimization, etc. Table 1 shows the performance test of cloud positioning of different models at different point sizes. In the point cloud of 0.9 MB, the optimized algorithm can reach 91 fps on an iPhone 13.

The redundancy of the point cloud size greatly affects the accuracy of the local point cloud positioning algorithm. Therefore, two local point cloud optimization solutions are designed: 1) Using filter feature descriptors to remove duplicates (Fig. 3); 2) Using test data to filter real and valid point cloud data and remove redundancy.

Considering that the positioning algorithm is an optimization problem (reducing a reprojection error), it is extremely affected by noise, and therefore a final obtained track is not smooth

▼**Table 1. Different mobile phone models in frames per second**

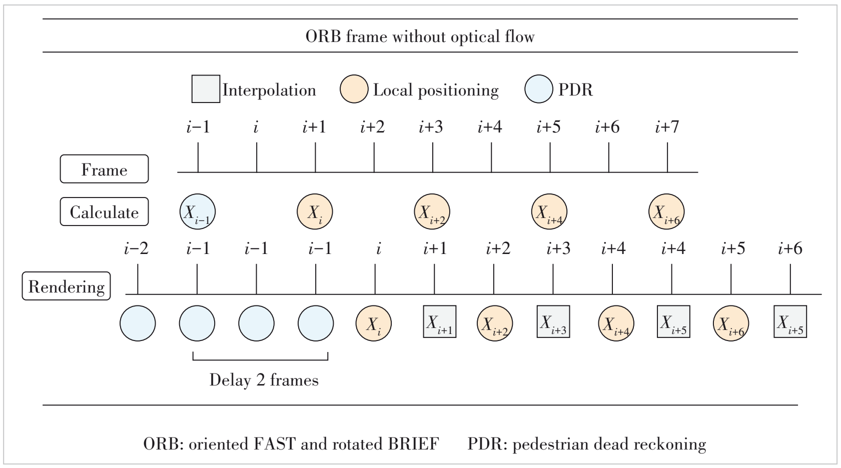
Descriptor Size/MB	Mobile Phone Model	Extracting ORB Features	Feature Matching (KNN)	PNP	Total Calculated Time/ms	Frames per Second/fps
0.9	MEIZU 11	1.052	38	4	50	20
	OnePlus 6	0.876	37	4	46	21
	Xiaomi 11	0.557	19	2	26	38
	Iphone 13	0.098	8	2	11	90

KNN: k-Nearest Neighbor ORB: oriented FAST and rotated BRIEF PNP: Perspective-n-Points enough. To ensure a stable anchoring effect, a filtering manner is used to optimize the positioning algorithm as follows.

1) A high-pass filter and a low-pass filter are used to eliminate incorrect positioning (Fig. 4);

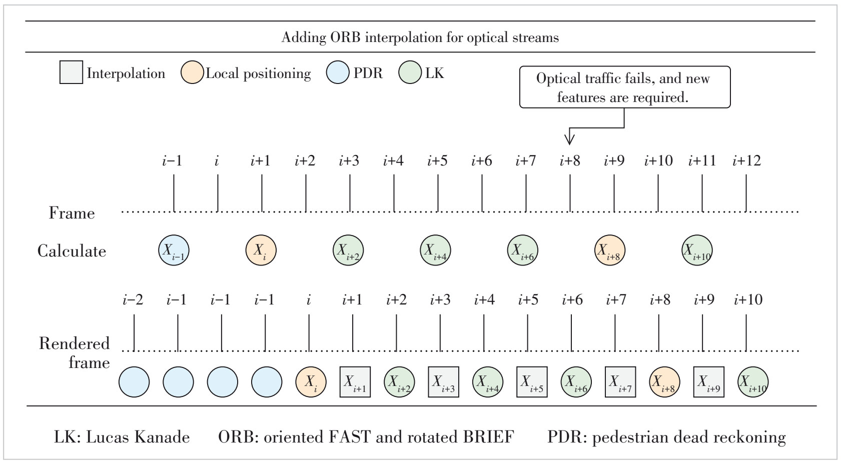
2) The camera position information of the first *k* frames and the sliding average value are used to smooth the track of the current frame.

Because the original environment on the web side supports only a single thread and the location algorithm based on the local point cloud cannot meet the real-time requirement (20 fps) with limited computing resources on the web side, the proposed algorithm is optimized by delaying video frames (Fig. 5). To ensure the stability of the local point cloud positioning algorithm, the optical flow tracking algorithm is introduced to improve the number of 2D-3D matches by using the previous frame’s prior knowledge, as shown in Fig. 6. Figs 7 and 8 show the experimental results without and with the optical flow respectively. The stability of model anchoring is improved during the positioning process.


ORB frame without optical flow
Interpolation
Local positioning
PDR
Frame
Calculate
Rendering
i−1 i i+1 i+2 i+3 i+4 i+5 i+6 i+7
X_{i-1} X_i X_{i+2} X_{i+4} X_{i+6}
i−2 i−1 i−1 i−1 i i+1 i+2 i+3 i+4 i+4 i+5 i+6
X_i X_{i+1} X_{i+2} X_{i+3} X_{i+4} X_{i+5} X_{i+6} X_{i+5}
Delay 2 frames

ORB: oriented FAST and rotated BRIEF PDR: pedestrian dead reckoning

▲**Figure 5. Delaying two frames without optical flow**

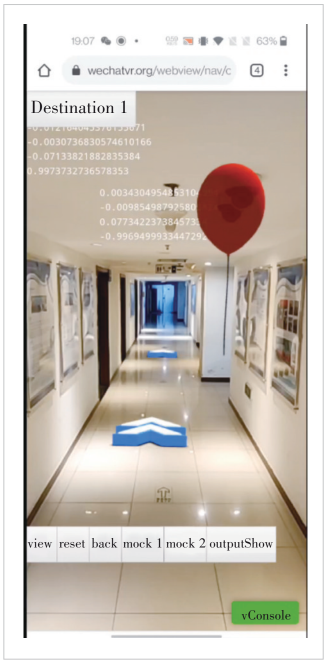

Adding ORB interpolation for optical streams
Interpolation
Local positioning
PDR
LK
Optical traffic fails, and new features are required.
Frame
Calculate
Rendered frame
i−1 i i+1 i+2 i+3 i+4 i+5 i+6 i+7 i+8 i+9 i+10 i+11 i+12
X_{i-1} X_i X_{i+2} X_{i+4} X_{i+6} X_{i+8} X_{i+10}
i−2 i−1 i−1 i−1 i i+1 i+2 i+3 i+4 i+5 i+6 i+7 i+8 i+9 i+10
X_i X_{i+1} X_{i+2} X_{i+3} X_{i+4} X_{i+5} X_{i+6} X_{i+7} X_{i+8} X_{i+9} X_{i+10}

LK: Lucas Kanade ORB: oriented FAST and rotated BRIEF PDR: pedestrian dead reckoning

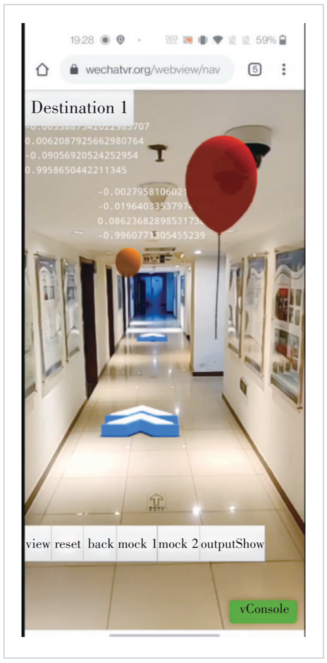
▲**Figure 6. Experiment of adding interpolation for optical streams**

4 Conclusions

This paper proposes a Web AR indoor navigation system based on local point cloud map positioning, which has beneficial effects on technical value compared with the prior art. It innovatively proposes the idea of point-cloud distribution, that is, to download the map of local point-cloud to the Web front end and use the computing power of the Web front end for positioning. Compared with an existing mainstream visual-based positioning manner for AR navigation, GPS-based positioning manner and multi-sensor fusion positioning manner, the positioning manner provided in the present invention does not depend on any additional sensor or external computing environment, thereby reducing development and deployment costs.

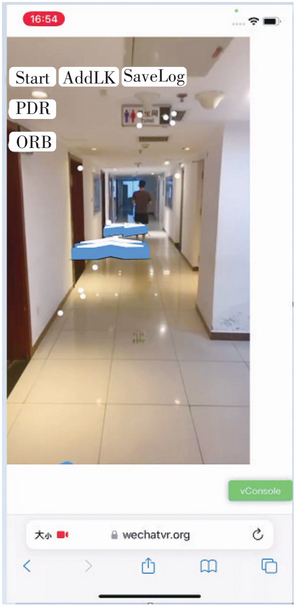

19:07
wechatvr.org/webview/nav/c
Destination 1
view reset back mock 1 mock 2 outputShow
vConsole

▲ **Figure 3. No filter is added for Web AR navigation**



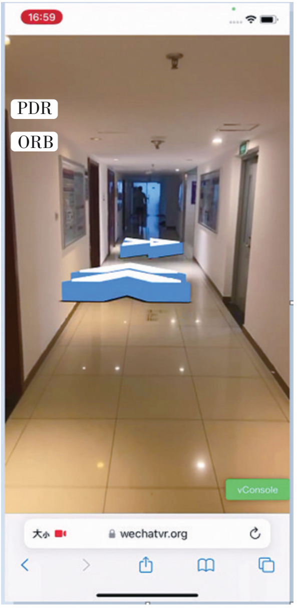
▲ **Figure 4. High-pass filter is added for Web AR navigation**

A lightweight web front-end positioning algorithm is presented for indoor Web AR navigation when the computational power of the web front-end is limited. A degree of dependence on network communication is reduced, the requirement of Web AR navigation on a network environment is reduced, and environment adaptability is improved. It takes a short time to deliver the web front-end positioning system to the point cloud for indoor Web AR navigation. In a small-scale indoor scenario, the positioning frequency on the iPhone 13 can reach 90 fps, which brings users a smooth user experience based on satisfying the real-time positioning requirements for Web AR navigation.



ORB: oriented FAST and rotated BRIEF
PDR: pedestrian dead reckoning

▲ **Figure 7. The proposed algorithm is optimized without optical flow**



ORB: oriented FAST and rotated BRIEF
PDR: pedestrian dead reckoning

▲ **Figure 8. Experiment diagram of Interpolation + Optical Flow**

Acknowledgement

We would like to acknowledge HUANG Yakun and LIU Jiarui for their contribution to this paper.

References

[1] CAO M W, JIA W, LV Z H, 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

[2] LINDENBERGER P, SARLIN P E, LARSSON V, et al. Pixel-perfect structure-from-motion with featuremetric refinement [C]//Proc. 2021 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2022: 5967 – 5977. DOI: 10.1109/ICCV48922.2021.00593

[3] YI K M, TRULLS E, LEPETIT V, et al. LIFT: learned Invariant Feature Transform [C]//European Conference on Computer Vision. Springer, 2016: 467 – 483. DOI: 10.1007/978-3-319-46466-4_28

[4] ZHANG X, YU F X, KARAMAN S, et al. Learning discriminative and transformation covariant local feature detectors [C]//Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017: 4923 – 4931. DOI: 10.1109/CVPR.2017.523

[5] ZHANG L G, RUSINKIEWICZ S. Learning to detect features in texture images [C]//Proc. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2018: 6325 – 6333. DOI: 10.1109/CVPR.2018.00662

[6] DETONE D, MALISIEWICZ T, RABINOVICH A. SuperPoint: self-supervised interest point detection and description [C]//Proc. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, 2018: 337 – 33712. DOI: 10.1109/CVPRW.2018.00060

[7] LAGUNA A B, RIBA E, PONSA D, et al. Key.Net: keypoint detection by handcrafted and learned CNN filters [C]//Proc. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 2020: 5835 – 5843. DOI: 10.1109/ICCV.2019.00593

[8] ONO Y, TRULLS E, FUA P, et al. LF-net: learning local features from images [C]//Proc. 32nd International Conference on Neural Information Processing Systems. NIPS, 2018: 6273 – 6247. DOI:10.5555/3327345.3327521

[9] SCHÖNBERGER J L, HARDMEIER H, SATTLER T, et al. Comparative evaluation of hand-crafted and learned local features [C]//Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017: 6959 – 6968. DOI: 10.1109/CVPR.2017.736

[10] WANG J, ZHOU F, WEN S L, et al. Deep metric learning with angular loss [C]//Proc. 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, 2017: 2612 – 2620. DOI: 10.1109/ICCV.2017.283

[11] ZAGORUYKO S, KOMODAKIS N. Learning to compare image patches via convolutional neural networks [C]//Proc. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2015: 4353 – 4361. DOI: 10.1109/CVPR.2015.7299064

[12] LUO Z X, SHEN T W, ZHOU L, et al. ContextDesc: local descriptor augmentation with cross-modality context [C]//Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020: 2522 – 2531. DOI: 10.1109/CVPR.2019.00263

[13] TIAN Y R, YU X, FAN B, et al. SOSNet: second order similarity regularization for local descriptor learning [C]//Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020: 11008 – 11017. DOI: 10.1109/CVPR.2019.01127

[14] SARLIN P E, DETONE D, MALISIEWICZ T, et al. SuperGlue: learning feature matching with graph neural networks [C]//Proc. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020: 4937 – 4946. DOI: 10.1109/CVPR42600.2020.00499

[15] SUN J M, SHEN Z H, WANG Y A, et al. LoFTR: detector-free local feature matching with transformers [C]//Proc. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2021: 8918 – 8927. DOI: 10.1109/CVPR46437.2021.00881

[16] SARLIN P E, CADENA C, SIEGWART R, et al. From coarse to fine: Robust hierarchical localization at large scale [C]//Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020: 12708 – 12717. DOI: 10.1109/CVPR.2019.01300

Biographies

SHI Wenzhe (shi.wenzhe@zte.com.cn) is a strategy planner and engineer for XRExplore Platform product planning at ZTE Corporation. He is also a member of the National Key Laboratory for Mobile Network and Mobile Multimedia Technology. His research interests include indoor visual AR navigation, SFM 3D reconstruction, visual SLAM, real-time cloud rendering, VR, and spatial perception.

LIU Yanbin is a strategy planner and product manager for XRExplore Platform product planning at ZTE Corporation. He is also a member of the National Key Laboratory for Mobile Network and Mobile Multimedia Technology. His research interests include real-time remote rendering, visual SLAM, and computer vision.

ZHOU Qinfen is the XR product leader director of new media industry and a senior architect of ZTE Corporation. She has more than 20 years of experience in the communication industry and media business. She has held the positions of product manager of short message center, product manager of cloud desktop, product line cost director, and XR product director at ZTE Corporation. She has a thorough understanding of products and related standards including Short Message Center, Cloud Desktop GPU Virtualization, XR, etc. As a member of the Shenzhen 8K UHD Video Industry Cooperation Alliance (SUCA) and Virtual Display Professional Committee of Jiangsu Communication Association, she leads a team responsible for the research on the latest video technology and related standards.