2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

CollabLoc: Collaborative Information Sharing for Real-Time Multiuser Visual Localization System

Teng-Te Yu[1], Yo-Chung Lau[2], Kai-Li Wang[1] and Kuan-Wen Chen[1*]

***Abstract*— This paper presents CollabLoc, a novel approach for real-time multi-user visual localization. Typically, localization systems employ a client-server design for locating cameras. In these systems, lightweight simultaneous localization and mapping computations are performed on the client side, while the server handles intensive localization tasks. This approach harnesses the complementary capabilities of the client and server, resulting in accurate, real-time localization results. However, existing architectures primarily operate on a one-to-one client-server structure, limiting their scalability and multi-user capabilities. Therefore, CollabLoc is designed to accommodate multiple clients through collaborative information sharing to considerably reduce computational overhead and enhance overall efficiency and accuracy. We propose a tracking confidence module that evaluates the tracking quality of individual clients and plays a pivotal role in prioritizing client requests by the server-side algorithm. On the server, we utilize fused poses to accelerate image retrieval. Moreover, we enhance the efficiency of optical flow estimation by employing a simplified feature extraction module and leveraging spatial similarities among neighboring clients to improve its performance. Finally, via the Pose Fusion Module, the server can periodically adjust fused poses to mitigate accumulated errors. Experimental results indicate that compared with a baseline method, CollabLoc improves positioning efficiency by nearly twice and achieves higher accuracy in multi-user scenarios.**

I. INTRODUCTION

Visual localization (VL) is a fundamental task in modern computer vision and robotics. Current VL methods excel under certain conditions but are still ineffective in specific scenarios. For example, simultaneous localization and mapping (SLAM) is fast but prone to cumulative errors that diminish accuracy with prolonged use. Conversely, single-shot localization (SSL) has high precision; however, its implementation is time-consuming and requires a prebuilt model.

To address this challenge, researchers have developed innovative approaches of combining the strengths of SLAM and SSL techniques. For example, the authors of [1] and [2] leveraged the characteristics of SLAM and SSL in a distributed setup with a client-server-based design in which lightweight SLAM algorithms on the client side performed real-time pose estimation. Key images were transmitted periodically to a server, which performed the computationally intensive SSL task of matching feature points extracted from images with a prebuilt feature point cloud to establish 2D-3D correspondences for precise pose estimation. The resulting pose was then returned to the client for pose fusion and cumulative error correction.

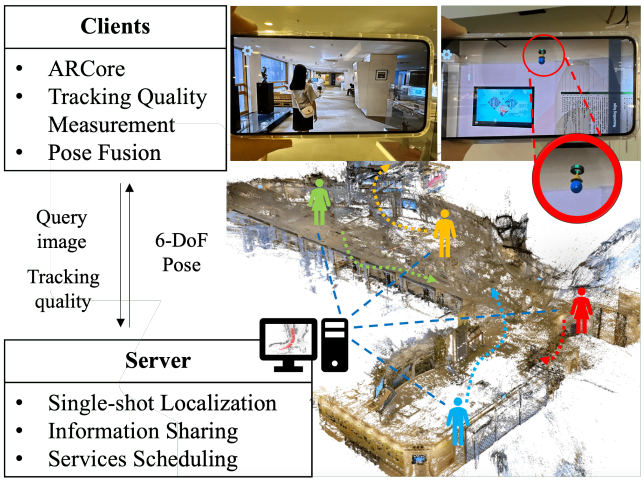

Fig. 1. Demonstration of our system used for a museum tour. The red circle highlights the virtual object that guides the user inside the museum.

The aforementioned client-server systems achieve VL by offloading heavy computations from the client to a server, thereby achieving real-time and accurate global localization. However, these methods are difficult to scale for multiple clients. Simultaneous multi-user localization is required for various VL applications, such as guided tours in museums [3], navigation in complex indoor spaces [4], or augmented-reality (AR) experiences [5]. In these applications, each added client either necessitates an additional server or linearly increases the computation time for client-server methods, which results in a poor user experience if extra servers are not available.

This paper introduces CollabLoc, an innovative multi-client VL system designed to overcome aforementioned limitations and enhance efficiency and accuracy through collaborative information sharing. To enhance the efficiency of multi-user real-time localization, we propose leveraging shared information between clients and the server. We introduce a tracking confidence module to assess client tracking quality, enabling a server-side scheduling algorithm to prioritize client service. Additionally, CollabLoc dynamically accelerates image retrieval and feature matching modules based on confidence values. Furthermore, the system enhances server-side localization efficiency by utilizing client-

[1]Teng-Te Yu, Kai-Li Wang and Kuan-Wen Chen are with the Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan. (E-mail of corresponding author* Kuan-Wen Chen: kuanwen@cs.nycu.edu.tw)

[2]Yo-Chung Lau is with Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 106, Taiwan and Digital Innovation Laboratory, Chunghwa Telecom Laboratories, Taoyuan 326, Taiwan.

side fused poses for optimized image retrieval, while simplifying optical flow estimation using a dedicated feature extractor module for adjacent pose transformations. Moreover, CollabLoc utilizes both temporal and spatial collaborative information to improve localization accuracy. Temporally, we adjust subsequent fused poses based on previous SSL poses to reduce accumulated errors in the pose fusion module. Spatially, we enrich the database by integrating client images via a database updater, enabling clients to share image information with others, thus facilitating easier access to similar viewpoint images. This enhancement consequently improves the performance of the optical flow network. Our experiments demonstrate that CollabLoc outperforms the *baseline* system (a modified version of [1]) in terms of efficiency and accuracy in multi-client scenarios. We validate our system through a museum tour navigation task (Fig. 1). The main contributions of this study are as follows:

- We developed CollabLoc, a VL system that collaboratively utilizes shared information to streamline computational processes, enhance overall efficiency, and improve localization accuracy for multiple users. To the best of our knowledge, no other client-server VL systems optimized for multi-user experiences have been developed.
- We developed a tracking confidence module for measuring client tracking quality for adaptive server-side prioritization and SSL acceleration.
- We improved server-side SSL efficiency by refining image retrieval with client-side fused poses. Moreover, we improved feature matching efficiency by leveraging optical flow with a simplified feature extractor module for adjacent pose transformations.
- We improved multi-user localization accuracy with a pose fusion module, adjusting fused poses based on SSL poses estimated by the server to minimize accumulated errors. Furthermore, we employed a database updater to add client images to the database, facilitating access to similar viewpoints and enhancing optical flow network performance.
- Compared with the *baseline* system, CollabLoc improves positioning efficiency by 1.91 times, along with higher efficiency and accuracy in multi-client scenarios.

II. RELATED WORK

A. Visual SSL

In SSL methods, pre-established models are used to estimate camera poses [6]. Structure-from-Motion (SfM) algorithms are often used for model construction by extracting feature points from scene photos and determining their corresponding 3D spatial coordinates. During the localization phase, the system extracts feature points from the query image and matches them with points in the prebuilt model, thereby establishing 2D-3D relations for the calculation of query image pose. This process is typically performed using the random sample consensus (RANSAC) [7] or Perspective-n-Point (PnP) algorithm. While hand-crafted features [8], [9], [10] have been employed in numerous VL systems, a recent study [11] has highlighted the limitations of such manually designed features. Consequently, more efficient techniques, such as NetVLAD [12] and SuperPoint [13], have been adopted for image retrieval and feature matching. Nevertheless, SSL methods require performing the complete process for each query image. By contrast, CollabLoc optimizes image retrieval by leveraging additional information in query images to identify mutually reusable relationships and reduce redundant computations.

B. Relative Pose Estimation

Visual SLAM [14], [15], [16], a type of localization and pose estimation method used for location tracking, is commonly employed in AR applications [17]. This is due to the capability of such methods to perform real-time localization while simultaneously constructing a 3D map of unknown environments. An end-to-end learning approach for this task was described in [18], emphasizing the importance of estimating relative camera poses between adjacent images. However, the learning-based pose estimation module developed in [18] exhibited lower accuracy compared to conventional geometric-based methods. The authors of [19] proposed using optical flow to enhance feature matching between adjacent images and applied the common 5-point algorithm [20] to calculate relative camera poses. CollabLoc also leverages optical flow networks to optimize feature matching and uses PnP algorithms to obtain precise camera poses efficiently.

C. Client-server VL System

Studies have adopted distributed approaches for allocating localization tasks by performing different techniques on separate devices in accordance with the device capabilities. For example, in [21], a client-server architecture was adopted for robot localization. In [1], fast Visual SLAM was implemented by the frontend, and key images were transmitted over the Internet to the backend as query images for SSL. A similar design was proposed in [2], but a place recognition module was incorporated to improve image retrieval accuracy by using sequential information. The aforementioned approaches are suitable for computationally limited devices, such as drones [22], because they offload intensive tasks to powerful devices. However, these client-server designs allocate computational tasks independently without fully integrating useful data; thus, client waiting time increases linearly for multi-user connections if additional servers are not used. On the other hand, the CollabLoc server efficiently harnesses client-side information to enhance efficiency and employs temporal and spatial information sharing to improve accuracy for multiple VL requests from various clients on a single server.

III. METHOD

Our CollabLoc system has a client-server architecture (Fig. 2); the frontend and backend components of this system communicate over a network connection. When the server

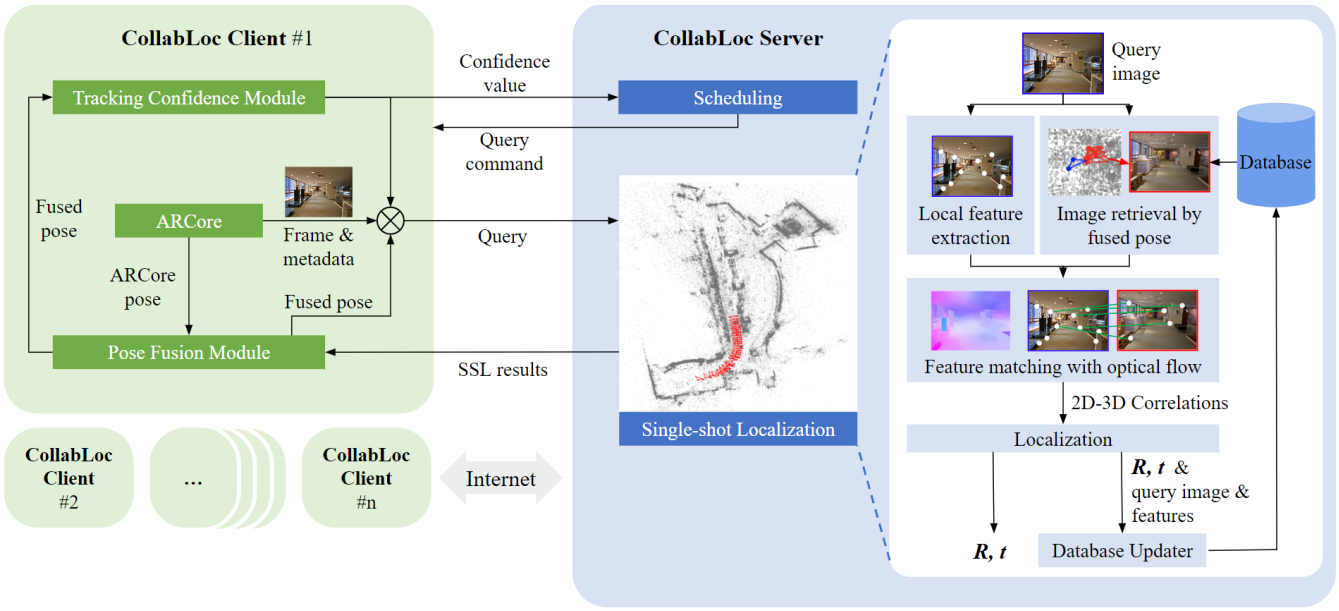

Fig. 2. Overview of the client-server data flow and server pipeline for *CollabLoc*.

receives a query image from a client, it performs SSL for the key camera frame and then returns the pose estimation results to the client. This SSL process is improved by revising image retrieval and feature matching on the basis of metadata within the query. Finally, the query image is added to the database through the *Database Updater*. At the client side, ARCore [1], introduced by Google in March 2018, is used to obtain the device's pose (30 fps). It employs SLAM technology to track the device's relative pose, known as the ARCore pose. The *Pose Fusion Module* then adjusts and refines the ARCore pose in real-time based on the received SSL result to generate a fused pose. Moreover, the client's *Tracking Confidence Module* evaluates the tracking quality, thereby aiding the server in client prioritization.

A. Tracking Confidence Module

Client tracking confidence is as an indicator of the quality of client fused pose measurements (i.e., tracking). This confidence value c is represented as a floating-point number ranging from 0 to 1; higher values indicate better client tracking. Tracking confidence is calculated at the client side by using the *Tracking Confidence Module* and is dependent on two key factors: the client's motion and the SSL filter. The client motion, accounting for the combined effects of translation and rotation, is calculated as a penalty for c, which is updated to c' as follows:

$$c' = c - s(P_{PrevSSL}, P_{Fused}), \tag{1}$$

$$\text{where } s(P_1, P_2) = dist(P_1, P_2) * \alpha_d + ang(P_1, P_2) * \alpha_a. \tag{2}$$

The approximate similarity value s quantifies the degree of similarity between two poses (i.e., the previous valid SSL pose $P_{PrevSSL}$ and current fused pose P_{Fused}). The function $dist(P_1, P_2)$ calculates the straight-line distance between two poses, while the function $ang(P_1, P_2)$ computes the angle. α_d and α_a are hyperparameters. The similarity value s is periodically calculated to decrease tracking confidence after movements or rotations.

The client compares the SSL pose P_{SSL} and corresponding fused pose P_{Fused} after receiving a result from the server. We define that if the distance difference between two poses exceeds 0.3 meters or the angle error is greater than 6 degrees, it indicates a significant disparity between them. In such cases, the SSL filter marks the SSL pose P_{SSL} as invalid, effectively filtering it out. Following that, the *Pose Fusion Module* continues to use the previous valid SSL pose $P_{PrevSSL}$ for pose fusion. The SSL filter is crucial for mitigating the effect of SSL errors; however, this filter might mistakenly identify an incorrect pose as valid, which results in it subsequently filtering out numerous correct poses. To ensure that erroneous pose identification does not result in an unrecoverable situation, the SSL filter is designed to affect the tracking confidence as follows:

$$c' = c + f(P_{SSL}, P_{Fused}), \tag{3}$$

where f outputs a value between -0.2 and 0.2 based on the poses; the smaller the distance and angle difference between the two poses, the higher the value f outputs, conversely, the lower the value f outputs. If c is below a predetermined threshold c_{min}, the client unconditionally adopts the next SSL result as a valid pose regardless of the SSL filter results and resets c to c_{reset}. This mechanism enables the client to use subsequent correct server poses for self-correction after adopting an erroneous SSL pose, thereby ensuring the accuracy and stability of the fused pose.

It's important to note that the tracking confidence c is

[1]https://developers.google.com/ar

updated every 0.1 seconds using Eq. (1). However, Eq. (3) is employed to update c only upon the client receiving an SSL pose. Following the update of tracking confidence to c' via either Eq. (1) or Eq. (3), if c' is less than 0, it is adjusted to 0; similarly, if c' exceeds 1, it is adjusted to 1. This guarantees that the value of c remains constrained within the range of 0 to 1.

The client periodically transmits its current tracking confidence value to the server. The server continuously receives the updated tracking status of multiple clients through a multithreaded architecture while performing SSL. After completing one round of the SSL process, the server prioritizes the client with the lowest c value received for the next round. This design is intended to equivalently maintain high tracking quality for multiple clients.

B. Pose Fusion Module

The *Pose Fusion Module* is tasked with converting the ARCore pose into the coordinate system of the SSL pose, thereby generating a global pose known as the fused pose. In cases where the client has not received an SSL pose yet, it utilizes the Rotation-Translation matrix between the previous valid SSL pose and the corresponding ARCore pose to convert the current ARCore pose into the fused pose. Upon receiving a new valid SSL pose, the client recalculates the Rotation-Translation matrix between the two poses and corrects subsequent fused poses accordingly. This module serves as a crucial link between the server and clients, leveraging temporal information sharing to correct subsequent fused poses based on previous SSL poses. In turn, this allows us to achieve rapid localization using SLAM while effectively suppressing accumulated errors, thereby enabling CollabLoc to achieve precise real-time localization. Furthermore, coupled with the accelerated server-side SSL method that we will introduce later, the frequency of correcting SLAM increases, effectively reducing the impact of accumulated errors.

C. Image Retrieval by Fused Pose

When the server sends a command to the selected client with the lowest c value, the client collects data and generates a query request comprising a camera frame, the client's fused pose, the intrinsic camera parameters, and c. The server then searches for images with similar poses to the fused pose in the prebuilt database by calculating the approximate similarity value s between the query image pose P_{Query} and all known poses P_i in the database as follows:

$$S = \{s(P_{Query}, P_i) \ \forall \ P_i \in DB\}, \tag{4}$$

where smaller values of S indicate superior matches. After sorting the values in S, a distributed retrieval approach extends the system's search to more distant locations after retrieving a certain number of suitable images. Images with higher similarity are prioritized for retrieval, while those with lower similarity may still be selected, but with reduced probability. This prevents the retrieval results from being affected by fused pose errors.

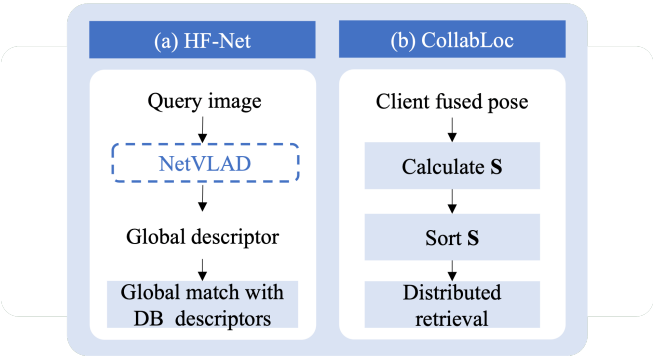

Fig. 3. Image retrieval pipelines of (a) HF-Net [11] and (b) CollabLoc.

The optimized pipeline efficiently leverages client-side information to extract images collaboratively, thereby resulting in a more efficient process. Experimental results demonstrate that using the HF-Net [11] pipeline for image retrieval takes 0.17247 seconds, while utilizing the CollabLoc pipeline reduces the image retrieval time to 0.02006 seconds, resulting in an increase in efficiency by 8.60 times. However, we retained the previous pipeline to enable the server to select a process on the basis of the client's tracking confidence value. Therefore, NetVLAD features can be used as a fallback if tracking is poor. Similarly, during the client's initial localization, the proposed system uses the original method for image retrieval because no fused pose is generated. Fig. 3 displays a comparison between the image retrieval pipelines of CollabLoc and HF-Net [11].

D. Feature Matching with Optical Flow

The hierarchical localization approach proposed in [11] involves matching the SuperPoint features extracted from a query image with those in the retrieved images for robustly establishing 2D-3D correspondences. However, this approach is still time-consuming. In the proposed system, a different strategy is adopted for the matching process; this process involves selecting the image I with the smallest s value from all retrieved images (i.e., I is the best retrieval result). The Recurrent All-Pairs Field Transforms (RAFT) [23] optical flow network is then used to compute the disparity between I and the query image for dense matching. Subsequently, the SuperPoint network is used to extract local features from the query image, and the positions are used as masks for the flow, thereby achieving sparse matching. This technique is similar to that in [19]; however, instead of computing the relative pose, RANSAC and PnP algorithms are used to identify 2D-3D correspondences and ultimately determine the pose. This approach facilitates increasing the number of correspondences to increase pose accuracy by adjusting the number of images processed by the flow network without the limitations of relative pose estimation.

Furthermore, the optical flow processing method is revised by pre-extracting features and context for images in the database by using the RAFT network. Thus, feature extraction is only performed for the query image during SSL, and the preprocessed features can be quickly accessed without

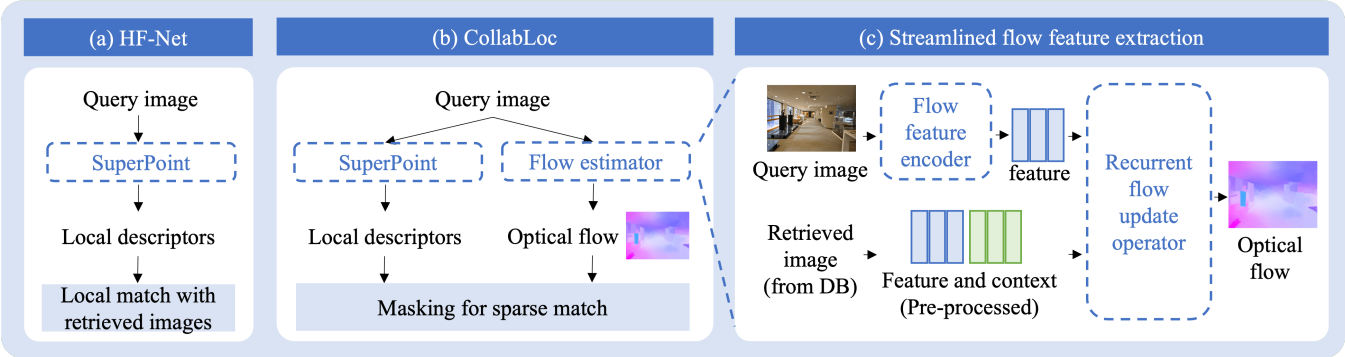

Fig. 4. Feature matching pipelines of (a) HF-Net [11] and (b) CollabLoc. (c) Streamlined flow of the feature extraction process displayed in (b).

frequent and repeated extraction. This adjustment improves efficiency, especially in multi-client situations.

Because the optical flow network requires that the view aspects of photos are similar, the original SuperPoint feature matching pipeline is retained as an alternative. The server selects a method based on the client's fused pose and tracking confidence. Specifically, when the client's tracking confidence value is greater than 0.4 and the retrieved image is sufficiently similar to the query image, the server chooses the optical flow pipeline. This ensures that the system can adapt to various localization scenarios. Fig. 4 reveals the differences between the two matching pipelines and presents our streamlined method for optical flow feature extraction.

E. Database Updater

After the server estimates the SSL pose, if the feature matching pipeline being used is HF-Net [11], indicating insufficient tracking confidence at the moment, the *Database Updater* takes responsibility for incorporating the query image, SSL pose, and features into the database. This makes it easier to retrieve images with similar viewpoints from the database during the subsequent SSL process for the same client. Similarly, for other clients whose proximity during the museum tour navigation task suggests comparable views, the server can access corresponding images during SSL processing. Later on, identifying similar viewpoint query and database images during image retrieval not only enhances the optical flow network's performance but also increases its activation frequency. The *Database Updater* allows clients to share photo information among themselves, enabling the server to find more suitable database photos during the image retrieval stage when processing other clients, including those that were not originally present in the database. Moreover, with more clients, encountering additional akin images leads to a larger collection in the database, providing us with more options when searching for images, thus enhancing localization performance. This underscores the benefits of sharing spatial similarity among clients.

IV. EXPERIMENTS

Implementation details. The server-side implementation using PyTorch [24] was based on our *baseline* architecture, which was a modified version of VB-GPS [1] that included HF-Net [11]. The *baseline* did not employ a mechanism for selecting appropriate retrieval and matching pipelines using pose priors, as this mechanism is a novel approach introduced by CollabLoc. The optical flow component employed the RAFT [23] architecture. During testing, the server had a GTX 2080 Ti graphics processing unit. The hyperparameters were as follows: $(\alpha_d, \alpha_a) = (0.005, 0.0003)$, $c_{min} = 0.2$, and $c_{reset} = 0.7$. The clients sent the tracking confidence value every 0.1 seconds.

Datasets. As there are no other large-scale image datasets with comprehensive ARCore reference information known to us, we utilize a self-created dataset as experimental material. The datasets were obtained by conducting SfM modeling for 525 images by using COLMAP [25], [26] for an indoor environment covering an area of approximately 44×35m^2 (Fig. 5). SuperPoint [13] features were then extracted from the images and triangulated using the pose of the model to obtain 3D coordinates. Moreover, NetVLAD [12] features and flow features from the images were pre-extracted and stored. We employed Android smartphones compatible with ARCore to capture trajectories comprising images and ARCore pose information within each scene at a rate of 5 fps. These trajectories, namely *track01* to *track12*, contained 259, 210, 212, 232, 205, 279, 264, 267, 250, 313, 206, and 235 images, respectively. Subsequently, clients establish communication with the server and retrieve the collected data to generate fused poses. We measured the computation time on the server and quantified the translation and rotation errors between the obtained fused poses and the ground truth poses that where obtained by adding the captured images to the prebuilt model and applying incremental SfM with COLMAP. Furthermore, we conducted on-site measurements to establish the scale of the COLMAP model in meters. The reported translation errors were calculated in meters on the basis of this scale.

A. Efficiency Evaluation on Single Client

We evaluate the single-client efficiency of each phase for the *baseline* and our CollabLoc pipeline, with both server each establishes a connection to a single client. Table I lists the average measured time (in seconds) for processing each image sent to server while tracking *track01*. Throughout the experiment, our image retrieval process was 2.64 times

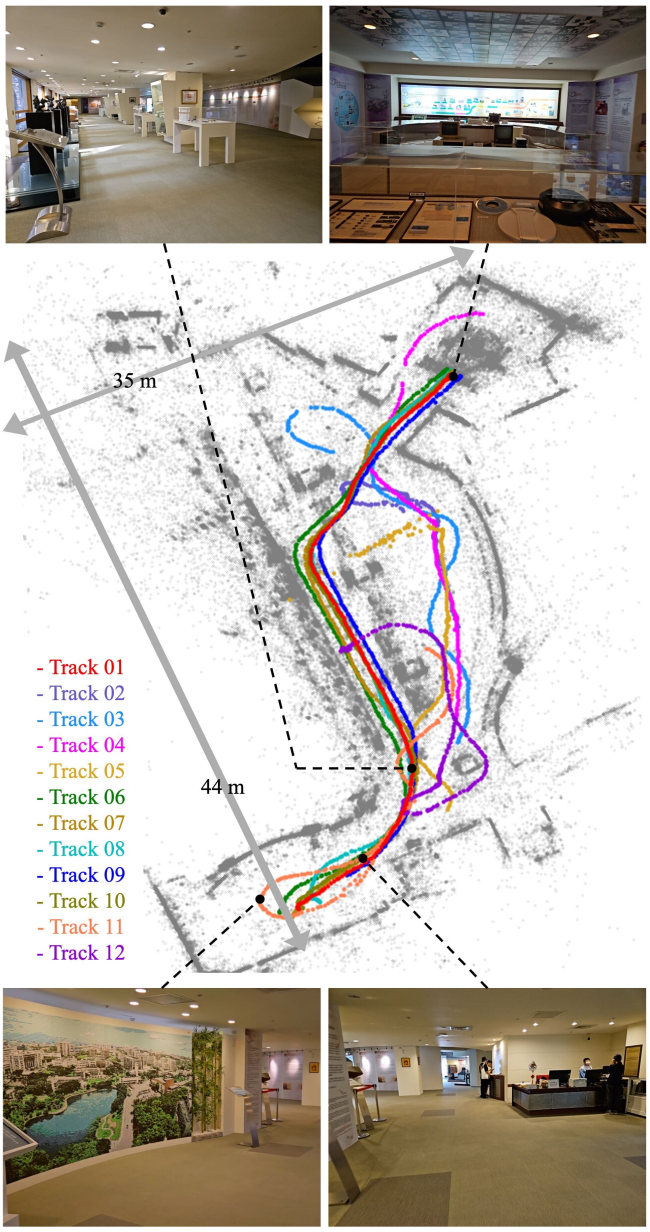

Fig. 5. Screenshots of our scene model, including the evaluated trajectories.

faster than that of the *baseline* system. This achievement is attributed to the server's effective use of the fused pose information received from the client, which enables it to avoid extracting and matching NetVLAD features. Moreover, the efficiency of the local feature matching phase increased by 2.20 times compared to that achieved using the original approach due to the utilization of optical methods in our method. The tracking confidence mechanism enabled feature point matching to be performed on only one image without compromising tracking performance. By contrast, matches from 20 images had to be calculated in the *baseline* method; thus, our pose calculation phase was over 3 times faster. Overall, our CollabLoc server was nearly twice as efficient as the *baseline* approach.

Table II lists the quantitative results for tracking *track01*

TABLE I

PIPELINE EFFICIENCY FOR BASELINE AND THE COLLABLOC SERVER

Phase	**Baseline** Time (s)	**Ours**	**Efficiency**
Local Feature Extraction	0.11118	**0.10849**	-
Image Retrieval	0.13428	**0.05094**	2.64x
Feature Matching	0.11625	**0.05277**	2.20x
Pose Calculation	0.11762	**0.03822**	3.08x
Total	0.47933	**0.25042**	1.91x

TABLE II

SINGLE-CLIENT EVALUATION

	Baseline		**Ours**	
	#	**Median Error (m / deg.)**	**#**	**Median Error (m / deg.)**
Fused	255	**0.06 / 0.74**	255	0.09 / 0.86
Available SSL	91	0.07 / 0.09	139	0.10 / 0.21
Filtered SSL	0	-	23	-

on both servers; median translation and rotation errors are reported in meters and degrees, respectively. Clearly, our server offered superior SSL services (i.e., the server provided clients with positioning updates more frequently). Although the SSL poses generated by our method might have lower accuracy than those generated by the *baseline* method for a single user, they are sufficient for indoor navigation or guidance applications. Our primary goal was to outperform traditional methods in terms of mitigating client errors in multi-user scenarios.

B. Qualitative Evaluation on Multi-client Performance

We implemented CollabLoc's scheduling algorithm on the *baseline* server since the orginal *baseline* method cannot handle multiple concurrent connections. We validated the effectiveness of our optimizations in terms of the pose accuracy for a multi-user scenario. Fig. 6 displays the localization results obtained for *track01* when five clients were simultaneously connected. More frequent SSL can enable clients to achieve more consistent motion trajectories, thereby enhancing the overall user experience. Furthermore, the translation error for fused poses tended to increase over time; however, the frequent server-side SSL updates mitigated these accumulated errors. A detailed localization process of CollabLoc can be viewed in https://youtu.be/jYZJpL2HhSw.

C. Quantitative Evaluation on Multi-client Performance

Table III presents the tracking performance for *track01* on a client with various numbers of users. When five client connections were used, CollabLoc had lower translation errors than did the *baseline* system, which indicated the effectiveness of the proactive server-side SSL approach for suppressing cumulative client errors. Furthermore, as the number of clients was increased, this effect became stronger. For twelve clients, the CollabLoc system achieved a fused pose error of only 13cm and thus was 1.69 times more accurate than the *baseline* system; the rotation error of CollabLoc was also lower. Thus, our multi-user design leverages collaborative data processing for effectively enhancing

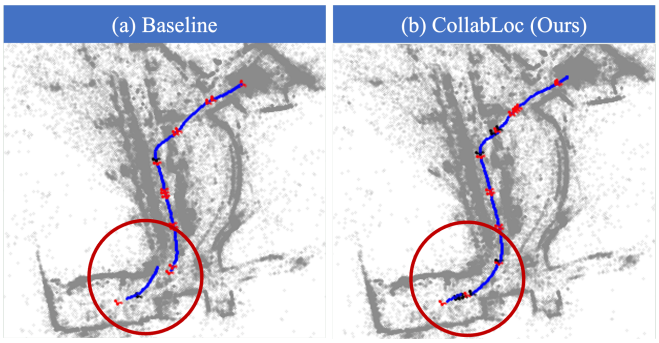

Fig. 6. Qualitative evaluation for a multi-client scenario. The circles highlight the greater smoothness of the CollabLoc trajectories compared with the *baseline* trajectories.

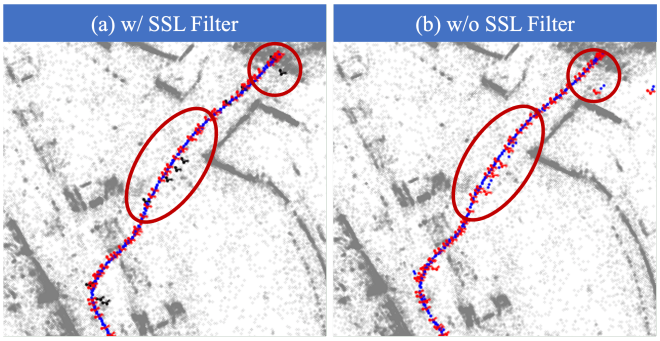

Fig. 7. Tracking results obtained for *track01* with and without the client-side SSL filter. Available SSL poses are presented in red, and filtered poses are presented in black.

TABLE III
MULTI-CLIENT EVALUATION

Clients		Baseline #	Baseline Median Error (m / deg.)	Ours #	Ours Median Error (m / deg.)
2	Fused	256	**0.07 / 0.77**	253	0.08 / **0.77**
	Available SSL	63	0.07 / 0.10	82	0.07 / 0.12
	Filtered SSL	1	-	2	-
5	Fused	249	0.11 / 0.92	257	**0.07 / 0.84**
	Available SSL	25	0.07 / 0.10	38	0.07 / 0.09
	Filtered SSL	4	-	1	-
8	Fused	251	0.13 / 0.92	257	**0.09 / 0.87**
	Available SSL	13	0.06 / 0.07	24	0.09 / 0.17
	Filtered SSL	2	-	3	-
12	Fused	215	0.22 / 0.95	247	**0.13 / 0.77**
	Available SSL	6	0.04 / 0.07	11	0.06 / 0.09
	Filtered SSL	1	-	3	-

TABLE IV
RESULTS OBTAINED WITH AND WITHOUT SSL FILTER

	w/ SSL Filter #	w/ SSL Filter Max Error (m / deg.)	w/o SSL Filter #	w/o SSL Filter Max Error (m / deg.)
Fused	255	**3.96 / 9.26**	257	6.70 / 23.88
Available SSL	139	**3.99 / 9.73**	161	6.72 / 24.41
Filtered SSL	23	2.83 / 8.68	-	-

processing efficiency and localization accuracy, particularly in scenarios involving numerous concurrent users.

D. Ablation Study

SSL filter plays a crucial role in tracking client-side activities. As illustrated in Fig. 7(a), a successful filtration process is vital for effectively preventing erroneous SSL information from disrupting the continuity of client tracking and contributes considerably to enhancing overall precision. As indicated in the Table IV, the SSL filter removed erroneous poses up to a distance of 2.83m, thereby ensuring that the fused pose remained accurate. Fig. 7(b) presents a tracking result obtained without the client-side SSL filter. In this case, all server results were considered valid during the tracking process. Consequently, SSL errors negatively affected client tracking and led to fragmented or discontinuous movement trajectories, thereby considerably reducing the user experience quality.

Images retrieved by fused pose are advantageous for estimating flow. Feature matching with optical flow can accelerate the pipeline, but the performance of this approach is only optimal for image pairs with similar view aspects. Fig. 8 depicts an example of the images retrieved by fused pose and NetVLAD [12] features. It is obvious that Fig. 8(b) and the query image share a closer viewpoint, demonstrating that utilizing the fused pose to retrieve images can be more beneficial for flow estimation. Furthermore, to avoid images having significantly different view angles that causes poor optical flow results, our server makes judicious use of the fused pose as a criterion when selecting the feature matching pipeline; it only employs optical flow for high-similarity image pairs, thereby ensuring more reliable matching results.

Database updater serves as a mechanism to boost the flexibility of the database, enabling CollabLoc to harness spatial similarity information shared among users, thereby enhancing localization performance, especially in scenarios with multiple nearby clients. Table V illustrates the localization results of *track01* with two concurrently connected clients, revealing that integrating the database updater mechanism improves the accuracy of both fused pose and

Fig. 8. An example of image retrieval: (a) query image and retrieved images by (b) our method and (c) NetVLAD [12].

TABLE V
RESULTS OBTAINED WITH AND WITHOUT DATABASE UPDATER

	w/ database updater #	w/ database updater Median Error (m / deg.)	w/o database updater #	w/o database updater Median Error (m / deg.)
Fused	253	**0.08 / 0.77**	255	0.11 / 0.84
Available SSL	82	**0.07 / 0.12**	59	0.10 / 0.14
Filtered SSL	2	-	11	-

SSL pose localization. Moreover, the frequency of utilizing the optical flow network is 96% in the version without the database updater, whereas it increases to 99% in the version with it. This suggests that employing the database updater can elevate the frequency of utilizing the optical flow network due to the availability of photos with similar perspectives in the database.

V. CONCLUSIONS

In this paper, we present CollabLoc, a real-time multi-user VL system that improves localization efficiency and accuracy through collaborative information sharing. Our approach utilizes shared information between clients and the server to enhance efficiency, integrating a tracking confidence module to evaluate client tracking quality and prioritize client service accordingly. Additionally, client-side fusion poses are employed to expedite image retrieval, while optical flow estimation is streamlined by simplifying feature extraction for faster 2D-3D correspondence matching. Furthermore, our method leverages both temporal and spatial collaborative information to enhance accuracy. It employs the pose fusion module to adjust subsequent fused poses based on previous SSL poses and enables clients to obtain image information from other clients using a database updater. Experimental results validate that our proposed method surpasses the *baseline* method in multi-user scenarios, achieving superior efficiency and accuracy while delivering a seamless multi-user experience in localization applications. As future work, we can gather optical flow maps from pre-calculated flow of nearby photos in the database. These maps can be used during SSL for new query images as a warm-start initialization for the RAFT network, reducing iteration counts to enhance efficiency.

ACKNOWLEDGMENT

This work was supported in part by Chunghwa Telecom Laboratories, Taoyuan, Taiwan and the National Science and Technology Council, Taiwan (111-2628-E-A49-003-MY2, 112-2634-F-A49-007-, and 113-2221-E-A49-164-MY3).

REFERENCES

[1] P.-J. Duh, Y.-C. Sung, L.-Y. F. Chiang, Y.-J. Chang, and K.-W. Chen, "V-eye: A vision-based navigation system for the visually impaired," *IEEE Transactions on Multimedia*, vol. 23, pp. 1567–1580, 2021.

[2] S. J. Lee, D. Kim, S. S. Hwang, and D. Lee, "Local to global: Efficient visual localization for a monocular camera," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2230–2239, 2021.

[3] D. Cai, "Museum navigation based on nfc localization approach and automatic guidance system," *International Journal of Computer Applications*, vol. 120, pp. 1–7, 2015.

[4] A. Blattner, Y. Vasilev, and B. Harriehausen-Mühlbauer, "Mobile indoor navigation assistance for mobility impaired people," *Procedia Manufacturing*, vol. 3, pp. 51–58, 2015. 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015.

[5] P.-E. Sarlin, M. Dusmanu, J. L. Schönberger, P. Speciale, L. Gruber, V. Larsson, O. Miksik, and M. Pollefeys, "LaMAR: Benchmarking Localization and Mapping for Augmented Reality," in *ECCV*, 2022.

[6] P. Ghosh, X. Liu, H. Qiu, M. A. M. Vieira, G. S. Sukhatme, and R. Govindan, "On localizing a camera from a single image," *ArXiv*, vol. abs/2003.10664, 2020.

[7] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, p. 381–395, jun 1981.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 91–110, Nov. 2004.

[9] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *2012 IEEE conference on computer vision and pattern recognition*, pp. 2911–2918, IEEE, 2012.

[10] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*, pp. 2564–2571, Ieee, 2011.

[11] P. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 12708–12717, IEEE Computer Society, jun 2019.

[12] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, 2018.

[13] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 337–33712, 2018.

[14] J.-C. Piao and S.-D. Kim, "Real-time visual–inertial slam based on adaptive keyframe selection for mobile ar applications," *IEEE Transactions on Multimedia*, vol. 21, no. 11, pp. 2827–2836, 2019.

[15] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pp. 225–234, IEEE, 2007.

[16] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[17] J. White, D. Schmidt, and M. Golparvar-Fard, "Applications of augmented reality," *Proceedings of the IEEE*, vol. 102, pp. 120–123, 02 2014.

[18] Y. Y. Jau, R. Zhu, H. Su, and M. Chandraker, "Deep keypoint-based camera pose estimation with geometric constraints," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4950–4957, 2020.

[19] J. Wang, Y. Zhong, Y. Dai, S. Birchfield, K. Zhang, N. Smolyanskiy, and H. Li, "Deep two-view structure-from-motion revisited," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 8949–8958, IEEE Computer Society, jun 2021.

[20] H. Li and R. Hartley, "Five-point motion estimation made easy," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 1, pp. 630–633, 2006.

[21] S. Ito, N. Kaneko, J. Takahashi, and K. Sumi, "Global localization from a single image in known indoor environments," in *2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pp. 70–75, 2018.

[22] K.-W. Chen, M.-R. Xie, Y.-M. Chen, T.-T. Chu, and Y.-B. Lin, "Dronetalk: An internet-of-things-based drone system for last-mile drone delivery," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15204–15217, 2022.

[23] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *Computer Vision – ECCV 2020* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), (Cham), pp. 402–419, Springer International Publishing, 2020.

[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.

[25] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.