2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

L-C*: Visual-inertial Loose Coupling for Resilient and Lightweight Direct Visual Localization

Shuji Oishi[1], Kenji Koide[1], Masashi Yokozuka[1], and Atsuhiko Banno[1]

***Abstract*— This study presents a framework, L-C*, for resilient and lightweight direct visual localization, employing a loosely coupled fusion of visual and inertial data. Unlike indirect methods, direct visual localization facilitates accurate pose estimation on general color three-dimensional maps that are not tailored for visual localization. However, it suffers from temporal localization failures and high computational costs for real-time applications. For long-term and real-time visual localization, we developed an L-C* that incorporates direct visual localization C* in a visual-inertial loose coupling. By capturing ego-motion via visual-inertial odometry to interpolate global pose estimates, the framework allows for a significant reduction in the frequency of demanding global localization, thereby facilitating lightweight but reliable visual localization. In addition, forming a closed loop that feeds the latest pose estimate to the visual localization component as an initial guess for the next pose inference renders the system highly robust. A quantitative evaluation of a simulation dataset demonstrated the accuracy and efficiency of the proposed framework. Experiments using smartphone sensors also demonstrated the robustness and resiliency of L-C* in real-world situations.**

I. INTRODUCTION

Visual localization (visual positioning system: VPS) is increasingly used in various applications, such as vehicle navigation for transportation and building inspection. Its easy setup is attractive in terms of the sensor cost and payload, resulting in application platforms ranging from autonomous systems to gaming services on smartphones. The principal algorithm infers an agile monocular camera pose in a given three-dimensional (3D) map from the camera view and can be divided into two types: indirect methods via feature point matching and direct methods via appearance comparison. Although recent methods of both types provide accurate and robust pose estimates [1] [2], direct methods have a significant advantage in terms of generality in that they operate on general color 3D maps that are not tailored for visual localization. Because city- or national-scale georeferenced 3D map data (point clouds and textured meshes) are commonly distributed, for instance, 3DCityDB [3] and CityGML 3.0 [4], direct methods play a central role in flexible localization systems for consumer devices. Despite its high demand, pure direct visual localization still suffers from temporal failure and high computational costs for real-time applications, which are serious problems in long-term localization.

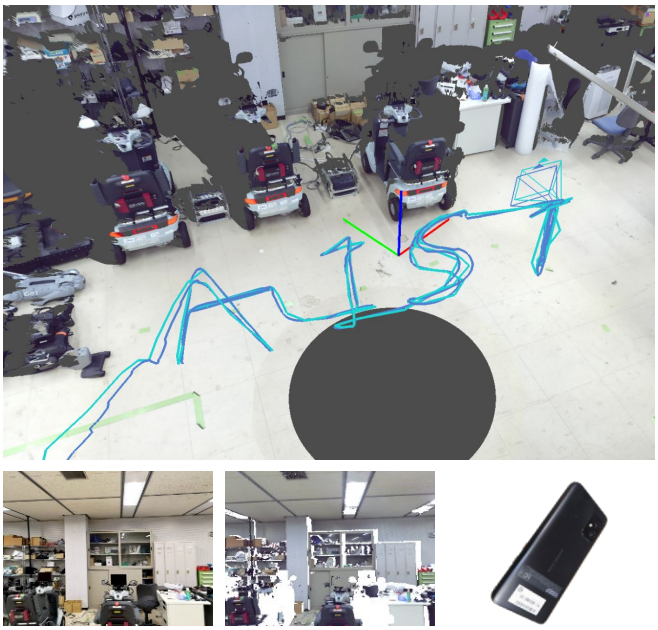


Fig. 1. 6-DoF global localization using a camera and an IMU embedded on a smartphone. Loose coupling of two types of pose estimates, low-rate direct global visual localization (C* on server side) and high-rate local ego-motion (ARCore on edge side), facilitates real-time and resilient long-term visual localization. See `https://youtu.be/1jdDb7_c1Ic` the attached video for more information.

The design of a multimodal localization framework that fuses complementary sensor observations is a promising solution for the ubiquitous system over usage and environments. This strategy is popular in state-of-the-art LiDAR SLAMs for avoiding geometric degeneracy [5] [6], and similar approaches can be used in visual localization for stable camera positioning [7]. Aiming toward a lightweight localization system, our interest here is the design of a sensor fusion architecture that requires less computational cost while retaining the localization quality, which facilitates mobile VPS on small computers.

In this study, we propose a sensor fusion framework for robust and lightweight visual localization. In a loose-coupling manner, the framework complementarily combines two pose factors: low-rate global poses and high-rate ego motions. Specifically, occasional global localization by C* [8] provides pose factors of accurate global pose estimates while visual-inertial odometry factors by VINS-Mono [9] or

*This work was supported by JSPS KAKENHI (Grant Number 22K12214) and a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

[1]Smart Mobility Research Team, National Institute of Advanced Industrial Science and Technology (AIST), Ibaraki, Japan {shuji.oishi, k.koide, yokotsuka-masashi, atsuhiko.banno}@aist.go.jp

ARCore[1] enable capturing of the local motions and track the latest pose in real-time. This relieves us of the frequent process of demanding visual localization, resulting in a lightweight system that is executable on mid- to low-end computers.

The main contributions of this study are as follows:

- A sensor fusion framework, *L-C**, for lightweight visual localization is proposed to enable 6-DoF full tracking of camera pose in a given 3D color map. While the complementary data fusion reduces the computational cost, forming a closed loop that feeds the latest pose estimate to the visual localization component as an initial guess for the next pose inference significantly makes the system robust.
- The architecture mainly comprises independent global and local tracking modules, and thus is suitable for edge-cloud computing, for instance, the case where ego-motion is estimated on an edge device while global pose is calculated on a server side.
- Detailed performance evaluations on simulation and real datasets are reported. The results reveal the advantages of L-C*; it enables maintenance of the localization accuracy even with low-rate global pose feeding while improving the resiliency in temporal failure of visual localization, resulting in stable and long-term visual localization compared with pure monocular localization.

II. RELATED WORK

Visual odometry / Visual SLAM Visual odometry and SLAM track the camera motion by simultaneously mapping a scene and localizing its pose. The map points are reconstructed via triangulation between pixels or feature points in adjacent frames, and reprojection of the "landmarks" allows estimation of the latest camera pose. By repeating this alternate process, these methods allow us to determine the $\mathbb{SE}(3)$ sensor motion in the coordinate system. Numerous sophisticated techniques have been developed, e.g., ORB-SLAM3 [1], OpenVINS [10], and DM-VIO [11], and apps are available on small computers or smartphones, such as ARCore and ARKit[2].

Visual localization Given a query image, visual localization infers a 6-DoF camera pose in a 3D map preconstructed using a camera. This process is equivalent to the front end of visual odometry / SLAM. Specifically, classical visual localization frameworks [12] first construct a feature map using visual SLAM or SfM with image descriptors. Next, they extract the same image features from the query image to perform 2D–3D association, and estimate the camera pose by minimizing the sum of the reprojection errors on the image plane. Although they provide lightweight visual localization, they operate only on feature maps. Direct methods facilitate the same function based on photometric errors and potentially operate on any color 3D maps. However, direct appearance comparison often suffers from lighting conditions or severe appearance changes. Some techniques have been proposed to overcome this problem [13] [14], as well as C* [8], which employs an information-theoretic metric for 6-DoF global localization in general photorealistic 3D maps. C* facilitates visual global localization that is highly accurate and robust to intensity variations between varying sensor properties and modalities. However, it requires relatively high graphics processing unit (GPU) computational power and can fail in ill-conditioned situations.

Visual-inertial fusion Sensor fusion is a promising approach for fail-safe systems. In the context of visual localization, fusing complementary observations from the vision sensor and IMU makes the system robust [7]. In this study, assuming various system setups, for instance, a single computer or an edge-cloud computing system, we developed a visual-inertial global localization framework L-C* in a loose-coupling manner, which allows module-wise system design, as described later.

Pose regression Another attractive approach for monocular camera localization is pose regression. Deep learning facilitates robust end-to-end pose estimation from camera images, and various methods have been proposed [15]. However, as reported in [16], neural pose regression tends to be less competitive than traditional visual localization frameworks in terms of accuracy. Recently, another approach, scene coordinate regression, has been extensively studied and has achieved state-of-the-art localization [17]. Despite their effectiveness, learning-based approaches require demanding preprocessing to collect a set of image samples with ground truth poses in the map, which may incur a causality dilemma. For ubiquitous systems, an easier setup is preferred for real-time and real-world applications.

III. PROPOSED METHOD

A. Notation

In the following description, a homogeneous transformation that transforms a 3D point from frame F to frame G is denoted by $\mathbf{T}_F^G \in \mathbb{SE}(3)$. The frames of the camera, IMU, and odometry at timestamp t_i are denoted by V_i, I_i, and O_i respectively, whereas the static world frame is denoted by W. Notably, the extrinsic parameters of frames V, I, and O are known via a calibration process and are assumed to be static. In addition, we define a sensor body frame B_i to represent all observations in a single coordinate system for simplicity. Thus, the state $\mathbf{x}_i$ to be estimated at timestamp t_i is expressed as follows:

$$\mathbf{x}_i = \mathbf{T}_W^{B_i} = [\mathbf{R}_i, \mathbf{v}_i], \tag{1}$$

where $\mathbf{R}_i \in SO(3)$ denotes the rotation matrix of the body frame in world coordinates and $\mathbf{v}_i \in \mathbb{R}^3$ denotes the translation. As explained later, we rely on an external module of visual-inertial odometry to extract the ego-motion factors. Thus, the biases of the accelerometer $\mathbf{b}_{acc_i}$ and gyroscope $\mathbf{b}_{\omega_i}$ of the IMU are not estimated explicitly in this framework. Notably, IMU preintegration [18] can be used as an alternative to odometry, which constrains the relative motion using only IMU data. In this case, the biases $\mathbf{b}_{acc_i}$, $\mathbf{b}_{\omega_i}$, and

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

[2]https://developer.apple.com/augmented-reality

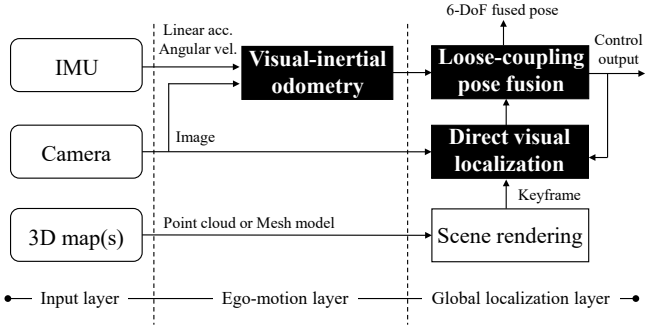


Fig. 2. Architecture of L-C*: Visual-inertial odometry module computes ego-motions based on incoming IMU and camera data, while the direct visual localization module localizes the global pose by comparing the view and the appearance of the given 3D map. Finally, the loose-coupling module fuses both estimates managing a factor graph and publishes a 6-DoF fused pose. Higher-rate control output is computed via short-term visual-inertial odometry for real-time applications and the pose feedback as the next initial guess of visual localization.

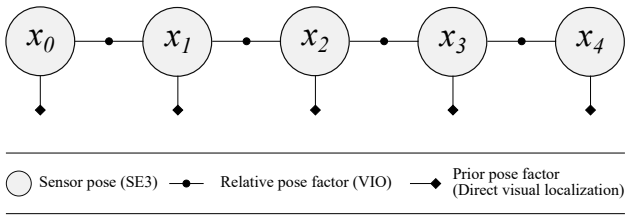


Fig. 3. Factor graph of our visual-inertial loose coupling for direct visual localization: Starting from the given initial pose, the graph incorporates global pose factors when provided by the direct visual localization. The relation between two global pose factors is determined by extracting ego-motion via visual-inertial odometry as a relative pose factor.

velocity $\mathbf{e}_i \in \mathbb{R}^3$ should be included in the state $\mathbf{x}_i$ for explicit optimization.

B. Factor graph

Fig. 2 illustrates an overview of the proposed fusion framework. Given an initial pose in a prior 3D map, the direct visual localization component begins to infer the current vision pose $\mathbf{T}_W^{V_i}$ by comparing the appearance of the 3D prior map with a scene image S_t. Simultaneously, the visual-inertial odometry component captures the local movement $\mathbf{T}_{V_{i-1}}^{V_i}$ at a higher rate. Generally, the odometry accumulates estimation errors over time, which incurs pose drift; however, the momentary motion estimate is assumed to be accurate. Because L-C* employs a sectional ego-motion estimate to bridge two global poses, it inherently enables drift-free global localization.

The flow of L-C* is expressed as a factor graph, as shown in Fig.3. To maintain and optimize the nonlinear factor graph, we use GTSAM library [19]. Specifically, the proposed framework relies on incremental smoothing and mapping (iSAM2) and variable elimination [20] for fixed-lag smoothing to ensure real-time processing. The details of the factors provided by the key blocks in Fig.2 are explained in the following subsections:

C. Direct visual localization for global pose estimation

Given a current image S_t, the direct visual localization block provides a prior factor for the global pose of the monocular camera in the 3D map. To perform localization in general photorealistic 3D maps that are not tailored to visual localization similar to feature maps in indirect methods, we employed our previous study C* [8], a direct localization method that evaluates the appearance similarity between the current camera view and 3D map. This facilitates highly accurate visual localization and is robust against changes in appearance / illumination.

Specifically, C* estimates the camera pose $\mathbf{T}_W^{V_i}$ using $\mathbb{SE}(3)$ local tracking against a synthetic key frame S_k rendered from a known viewpoint $\mathbf{T}_W^K$ for efficient localization. Because the keyframe comprises a color image S_k and depth map D_k, it can be regarded as a submap of the provided 3D map, and a color 3D point ${}_i\mathbf{P}^k$ is reconstructed from each pixel ${}_i\mathbf{u} = ({}_iu, {}_iv) \in S_k$. Given the relative pose $\mathbf{T}_K^{V_i}$, ${}_i\mathbf{P}^K$ can be reprojected to the current image plane as ${}_i\mathbf{u}' = \pi(\mathbf{T}_K^{V_i} \cdot {}_i\mathbf{P}^K)$, where $\pi : \mathbb{R}^3 \mapsto \mathbb{R}^2$ denotes the camera projection model with the known intrinsic parameters. This process enables us to find the pixel-wise correspondences between the current image S_t and keyframe image S_k, and the optimal relative pose $\hat{\mathbf{T}}_K^{V_i}$ is obtained by minimizing the sum of the per-pixel differences. For a robust appearance comparison, C* leverages the normalized information distance (NID) [21], and the cost function to be minimized is defined as follows:

$$\hat{\mathbf{T}}_K^{V_i} = \underset{\mathbf{T}_K^{V_i}}{\arg\min} \delta I_{NID}\left(S_t, S_k, \mathbf{T}_K^{V_i}\right), \tag{2}$$

$$\delta I_{NID}\left(S_t, S_k, \mathbf{T}_K^{V_i}\right) \equiv \frac{H_{t,k}\left(\mathbf{T}_K^{V_i}\right) - I_{t,k}\left(\mathbf{T}_K^{V_i}\right)}{H_{t,k}\left(\mathbf{T}_K^{V_i}\right)}. \tag{3}$$

where $H_{t,k}$ and $I_{t,k}$ denote the joint entropy and mutual information, respectively, and are calculated based on the color co-occurrence between S_t and S_k as follows:

$$\mathrm{H}(S_t) = -\sum_{x=1}^{n} p_t(x)\log\left(p_t(x)\right), \tag{4}$$

$$\mathrm{H}(S_t, S_k) = -\sum_{x=1}^{n}\sum_{y=1}^{n} p_{t,k}(x,y)\log\left(p_{t,k}(x,y)\right), \tag{5}$$

$$I(S_t; S_k) = H(S_t) + H(S_k) - H(S_t, S_k), \tag{6}$$

where $p_{t,k}$ denotes the joint probability of an $n \times n$-dimensional histogram, and the marginal probabilities p_t and p_k are derived from $p_{t,k}$.

The Broyden–Fletcher–Goldfarb–Shannon (BFGS) algorithm was employed to determine the optimal relative pose $\mathbf{T}_K^{V_i}$. Starting from a given initial guess or a previous estimate, BFGS iteratively solves Eq.2 using the Jacobian of the NID.

$$\begin{aligned} {}^{(i+1)}\mathbf{T}_K^{V_i} &= {}^{(i)}\mathbf{T}_K^{V_i} - \alpha B_k^{-1}\frac{d\delta I_{NID}}{d^{(i)}\mathbf{T}_K^{V_i}}, \\ \frac{d\delta I_{NID}}{d\mathbf{T}_K^{V_i}} &= \frac{\left(\frac{dH_{t,k}}{d\mathbf{T}_K^{V_i}} - \frac{dI_{t,k}}{d\mathbf{T}_K^{V_i}}\right)H_{t,k} - \left(H_{t,k} - I_{t;k}\right)\frac{dH_{t,k}}{d\mathbf{T}_K^{V_i}}}{H_{t,k}^2}. \end{aligned} \tag{7}$$

Based on the optimal relative pose $\hat{\mathbf{T}}_K^{V_i}$, we obtain the global pose of the vision as $\mathbf{T}_W^{V_i} = \hat{\mathbf{T}}_K^{V_i} \circ \mathbf{T}_W^K$ and feed it to the nonlinear factor graph as a new $\mathbb{SE}(3)$ global pose factor.

D. Visual-inertial odometry for ego-motion extraction

To extract the factor of local motion bridging camera poses at two time points, we employ visual-inertial odometry estimation. Visual odometry / visual SLAM allows us to track camera motion by simultaneously reconstructing an environmental map and localizing its pose via scene cloud reprojection. Because it only provides Sim3 pose of the sensor with an undetermined scale, we used visual–inertial odometry / SLAM to obtain the motion $\mathbf{T}_W^{O_i} \in \mathbb{SE}(3)$. Specifically, in the evaluation and experiments described in Section IV, we employed VINS-Mono [9] and AR core for odometry measurements.

To bridge successive global poses, the framework extracts relative motion during the time span between poses using odometry. Visual-inertial odometry provides sequential poses in the odometry coordinate system at a higher rate. When the global pose factors arrive at timestamps t_{i-1} and t_i, the relative transformation factor between states $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ can be extracted as follows:

$$\mathbf{T}_{B_{i-1}}^{B_i} = \mathbf{T}_{O_i}^{B_i} * \mathbf{T}_W^{O_i} * \mathbf{T}_{O_{i-1}}^{W} * \mathbf{T}_{B_{i-1}}^{O_{i-1}}. \tag{8}$$

Note that the transformation from body frame to odometry frame $\mathbf{T}_{O_i}^{B_i}$ is assumed to be static in our impelentation.

E. Control output and closed loop

The proposed framework provides fused pose outputs at the maximum rate of visual localization. The frequency of visual localization can be significantly low (several hertz); however, depending on the application, higher-rate outputs are required, for instance, real-time robot control. To obtain the control output, we estimated the latest camera pose by calculating the short-term visual-inertial odometry starting from the last fused pose, as shown in Fig.2. As mentioned in III-A, IMU dynamics-based motion prediction is also useful for capturing momentary motion at much higher IMU rates, for instance, several hundred Hertz.

This pose "prediction" significantly stabilizes the visual localization component. C* [8] localizes a monocular camera by repeating keyframe-based local tracking, and every iteration of the pose update begins from the last estimate. Because the NID metric used in the algorithm has an extremely small convergence basin, a poor initial guess that is far from the optimal value often leads to localization failure. Thus, to maintain the health of the localizer, the frequency should be maintained at the maximum, which results in accurate but demanding visual localization. However, reliable pose prediction can overcome the drawback by forming a closed loop (Fig.2) that feeds the predicted pose into C* as the initial guess of the next optimization. This significantly stabilizes the visual localization even in low-rates, which further makes the entire system resilient to temporal failure of visual localization and prevents an loss of the localizer.

IV. EXPERIMENTS

In this section, the performance of the proposed framework is examined from several perspectives. First, we conducted tryouts on the simulation data to quantitatively evaluate the performance of L-C*. For comparison, the localization results from our previous study C* [8] were also evaluated, which revealed the accuracy, robustness, and efficiency of the loose coupling of different observations against monocular camera localization. We also demonstrated the capability of localization in real situations, revealing its usability and applicability.

A. Setup

Dataset: For quantitative evaluation, we used the Replica Dataset [22] that provides a set of photorealistic color 3D models of different rooms. We defined random trajectories in 3D models and generated sequences of monocular camera images for localization. IMU data were synthesized simultaneously by referring to the spec of ADIS16448 (Isensor Co., Ltd.) used in EuRoC dataset to calculate visual-inertial odometry.

Next, 3D mesh maps were constructed for real-world demonstrations. To construct 3D mesh models of real environments, Focus3D (FARO Technologies, Inc.), which enables the capture of highly-dense colored 3D points, was used. The spherical scan also allowed us to generate triangular meshes by simply connecting adjacent points with a certain threshold length. Camera and IMU data sequences were captured using a Zenfone 8 (ASUSTek Computer, Inc.) while moving dynamically.

Computation The visual localization component (C*) relies heavily on GPU processing; thus, we used a laptop PC with an NVIDIA GeForce RTX 2070 Super for subsequent evaluations and experiments. Notably, although C* requires relatively high GPU power, it runs faster than 30 Hz against the VGA video stream on the graphics card. L-C* further reduces the entire computational cost of visual localization, which enables various system setups including edge clouds, remote processing, and stand-alone computing, depending on the machine specs (Fig.4). Specifically, quantitative evaluation was performed using a laptop (Fig.4(c)) while we used a Zenfone 8 and the laptop for edge- and server-side computing (Fig.4(a)) in the live demonstration.

B. Quantitative evaluation in the Replica Dataset

To evaluate the performance of L-C* quantitatively, we used two photorealistic 3D models from the Replica Dataset [22] (Fig.5). Specifically, we selected "apartment_0" and "frl_apartment_1" that are the largest models in the dataset for longer trajectory generation including both translational and rotational motions. As explained in IV-A, the test image sequences were synthesized along predefined trajectories in the corresponding 3D models. To further test the resiliency in temporal localization failure, we prepared a test image sequence in "frl_apartment_2", where pieces of the furniture were rearranged from frl_apartment_1, and performed the

TABLE I

LOCALIZATION RESULTS ON REPLICA DATASET: ABSOLUTE TRAJECTORY ERRORS [M] AND RELATIVE TRAJECTORY ERRORS [M] (PER 10 M) ARE SHOWN.

Map	Image Hz	C* [8]		L-C* /wo loop		L-C* /w loop (proposed)	
		ATE [m]	RTE [m]	ATE [m]	RTE [m]	ATE [m]	RTE [m]
apartment_0	30	**0.0030 ± 0.0015**	**0.0072 ± 0.0032**	0.0034 ± 0.0016	0.0080 ± 0.0041	0.0034 ± 0.0016	0.0080 ± 0.0041
	10	**0.0039 ± 0.0027**	**0.0092 ± 0.0058**	0.0061 ± 0.0033	0.011 ± 0.0052	0.0060 ± 0.0033	0.011 ± 0.0051
	3	1.8 ± 0.92	2.5 ± 1.6	1.8 ± 0.90	2.2 ± 1.4	**0.016 ± 0.0097**	**0.024 ± 0.012**
	1	2.2 ± 0.75	3.8 ± 1.6	2.1 ± 0.65	3.8 ± 1.5	**0.044 ± 0.029**	**0.10 ± 0.081**
frl_apartment_1†	30	0.87 ± 0.63	1.7 ± 1.4	0.87 ± 0.62	1.8 ± 1.4	**0.084 ± 0.065**	**0.13 ± 0.096**
	10	2.4 ± 1.2	4.7 ± 1.8	2.4 ± 1.2	4.1 ± 1.2	**0.055 ± 0.038**	**0.09 ± 0.059**
	3	2.3 ± 0.90	4.2 ± 1.5	2.3 ± 0.93	4.3 ± 1.3	**0.074 ± 0.048**	**0.11 ± 0.071**
	1	2.4 ± 1.1	-	2.3 ± 1.1	3.4 ± 0.93	**0.14 ± 0.074**	**0.25 ± 0.14**

†: The image sequence is captured in frl_apartment_2 where pieces of the furniture are rearranged from frl_apartment_1 to examine the robustness against appearance changes.

TABLE II

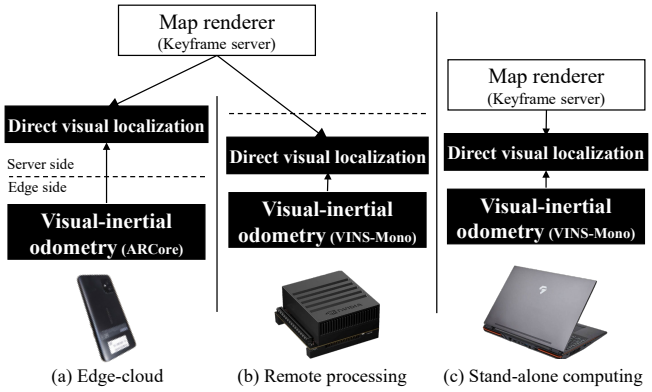
LOCALIZATION SUCCESS RATE ON REPLICA DATASET (THE VALID TRACKING DURATION UNTIL ATE EXCEEDS 1.0 [M]) [%] .

Map	Image Hz	Success rate [%]		
		C* [8]	L-C* /wo loop	L-C* /w loop (proposed)
apartment_0	30	**100**	**100**	**100**
	10	**100**	**100**	**100**
	3	3.00	20.6	**100**
	1	0	0	**100**
frl_apartment_1	30	60.5	58.2	**100**
	10	0	0	**100**
	3	0	0	**100**
	1	0	0	**100**

TABLE III

LOCALIZATION SUCCESS RATE ON REAL DATASET (THE DURATION OF VALID TRACKING) [%] .

Map	Image Hz	Success rate [%]		
		C* [8]	L-C* /wo loop	L-C* /w loop (proposed)
indoor	30	58.8	65.4	**100**
	10	14.5	18.2	**100**
	3	14.2	17.3	**100**
	1	13.3	13.3	**100**
outdoor	30	9.71	10.0	**100**
	10	6.69	8.53	**100**
	3	5.35	6.02	**100**
	1	6.02	7.70	**100**



(a) Edge-cloud (b) Remote processing (c) Stand-alone computing

Fig. 4. Different system setup: Lightweight visual localization by L-C* enables a variety of computing styles, e.g., edge-cloud, remote processing, and stand-alone computing.

localization tryout in frl_apartment_1 to examine the robustness against appearance changes. Starting from an initial guess manually provided via our OpenGL viewer, the camera pose of each test image was estimated using the proposed method L-C*. We also tested C* [8] and L-C* without a feedback loop as competitors to demonstrate the validity of the proposed framework.

Table I lists the results of the quantitative evaluation. The estimated trajectories were evaluated as absolute trajectory error (ATE) [m] and relative trajectory error (RTE) [m] per 10m. All the methods successfully estimated accurate camera poses in apartment_0 when the image stream was fed at high rates (30 and 10 Hz). However, when the frequency decreased, only L-C* could track the agile camera while maintaining high accuracy against the ground truth. As summarized in Table II, the other methods were immediately lost owing to a lack of camera observations.

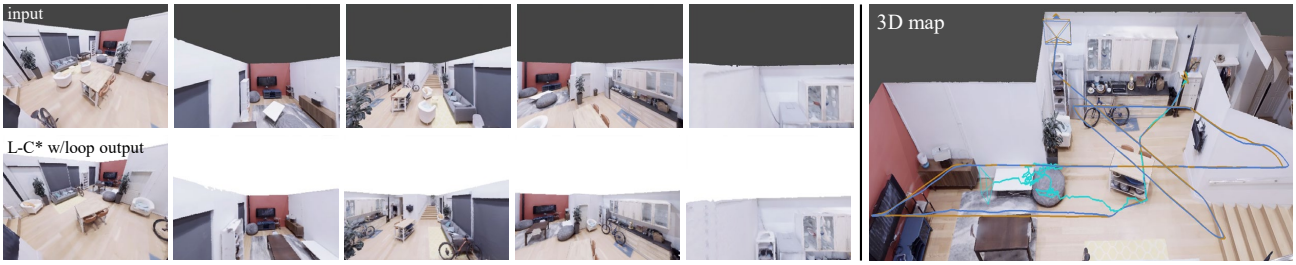
The results for frl_apartment agree with those for apartment_0. In this evaluation, severe appearance changes worsened the localization performance, and pure C* and L-C* without the feedback loop failed in all trials. L-C*, however, accurately localized the monocular camera, even in cases of appearance changes, by employing loose coupling, thereby outperforming its competitors.

C. Demonstration in the real world

To demonstrate the effectiveness of the proposed framework in the real world, we present the results on a dataset captured using Focus3D for 3D maps and Zenfone 8 for sequential camera and IMU data. Fig. 6 shows the image sequences and 3D maps. As depicted in Fig. 4(a), odometry estimation operates on the smartphone, and visual localization against camera images runs on a laptop with a GeForce RTX2070 Super, which simulates edge-cloud computing.

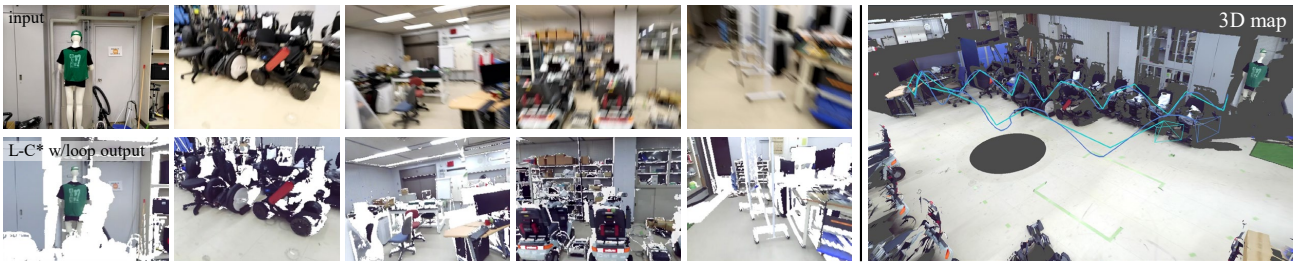
The localization success rate and duration of valid localization determined manually are listed in Table III. Owing to dynamic motion or 3D map sparsity, competitors could not track the agile camera completely. Moreover, as the frequency of the camera stream decreases, the localization success rate decreases significantly. Even in severe situations, L-C* robustly estimated the camera pose and accomplished all trials, demonstrating the effectiveness of loose coupling of direct visual localization and visual-inertial odometry.

(a) Replica Dataset: apartment_0



(b) Replica Dataset: frl_apartment_1

Fig. 5. Quantitative evaluation on Replica dataset (Orange: Groud truth, Green: C*, Blue: L-C* in the case of 3 Hz image feeding): The image and IMU data sequences were captured along pre-defined trajectories.



(a) Indoor environment

(b) Outdoor environment

Fig. 6. Demonstration on dataset captured in the real world (Green: 3 Hz global localization, Blue: Fused pose (control output)).

V. CONCLUSIONS

We propose a sensor fusion framework for a lightweight visual localization system. The loose coupling of different pose factors, occasional global pose estimates and frequent local ego-motions, makes the entire system resilient to temporal absence and failure of visual localization. The evaluations demonstrated that the framework achieved reliable and robust visual localization while reducing the high computational cost of frequent global pose estimation. This architectural advantage inherently allows us to run a localization module on a mid- to low-end computer that simultaneously boosts distributed systems.

As demonstrated in Section IV, L-C* allows 6-DoF full tracking with a minimum frequency for demanding global localization. Because this significantly reduces the computational cost of direct visual localization, we intend to implement the entire system on a small PC without GPU cards or smartphone to facilitate easy localization on a lightweight consumer device. We also used the framework in various applications, such as autonomous navigation of personal mobility vehicles, egocentric action recognition, and human—robot interaction. We are also interested in adaptation to each situation, and other factors, such as the wheel encoder or human intention, may be useful information to further constrain the camera pose [23]. Thus, we believe that the adaptive extension of the factor graph would be an interesting topic.

REFERENCES

[1] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[2] K. Ok, W. N. Greene, and N. Roy, "Simultaneous tracking and rendering: Real-time monocular localization for MAVs," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 4522–4529.

[3] Z. Yao, C. Nagel, F. Kunde, G. Hudra, P. Willkomm, A. Donaubauer, T. Adolphi, and T. H. Kolbe, "3DCityDB - a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML," *Open Geospatial Data, Software and Standards*, vol. 3, no. 1, 2018.

[4] T. Kutzner, K. Chaturvedi, and T. H. Kolbe, "CityGML 3.0: New functions open up new applications," *Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 88, no. 1, pp. 43–61, 2020.

[5] S. Khattak, H. Nguyen, F. Mascarich, T. Dang, and K. Alexis, "Complementary multi–modal sensor fusion for resilient robot pose estimation in subterranean environments," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2020, pp. 1024–1029.

[6] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Globally consistent and tightly coupled 3d lidar inertial mapping," in *IEEE International Conference on Robotics and Automation (ICRA2022)*, May 2022, pp. 5622–5628.

[7] K. Qiu, T. Liu, and S. Shen, "Model-Based Global Localization for Aerial Robots Using Edge Alignment," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1256–1263, 2017.

[8] S. Oishi, Y. Kawamata, M. Yokozuka, K. Koide, A. Banno, and J. Miura, "C*: Cross-modal simultaneous tracking and rendering for 6-dof monocular camera localization beyond modalities," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5229–5236, 2020.

[9] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[10] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A research platform for visual-inertial estimation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Paris, France, 2020. [Online]. Available: \url{https://github.com/rpng/open_vins}

[11] L. von Stumberg and D. Cremers, "DM-VIO: Delayed marginalization visual-inertial odometry," *IEEE Robotics and Automation Letters (RA-L) & International Conference on Robotics and Automation (ICRA)*, vol. 7, no. 2, pp. 1408–1415, 2022.

[12] M. Donoser and D. Schmalstieg, "Discriminative feature-to-point matching in image-based localization," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 516–523.

[13] T. Caselitz, B. Steder, M. Ruhnke, and W. Burgard, "Monocular camera localization in 3D LiDAR maps," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1926–1931.

[14] H. Yu, W. Zhen, W. Yang, J. Zhang, and S. Scherer, "Monocular camera localization in prior lidar maps with 2d-3d line correspondences," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4588–4594.

[15] F. Xue, X. Wu, S. Cai, and J. Wang, "Learning multi-view camera relocalization with graph neural networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 372–11 381.

[16] Q. Zhou, T. Sattler, M. Pollefeys, and L. Leal-Taixé, "To Learn or Not to Learn: Visual Localization from Essential Matrices," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[17] Z. Huang, H. Zhou, Y. Li, B. Yang, Y. Xu, X. Zhou, H. Bao, G. Zhang, and H. Li, "VS-Net: Voting with segmentation for visual localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6101–6111.

[18] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transaction on Robotics*, vol. 33, no. 1, p. 1–21, 2017.

[19] F. Dellaert and G. Contributors, "borglab/gtsam," May 2022. [Online]. Available: https://github.com/borglab/gtsam)

[20] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3281–3288.

[21] M. Li, X. Chen, X. Li, B. Ma, and P. M. B. Vitanyi, "The similarity metric," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3250–3264, Dec 2004.

[22] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, A. Clarkson, M. Yan, B. Budge, Y. Yan, X. Pan, J. Yon, Y. Zou, K. Leon, N. Carter, J. Briales, T. Gillingham, E. Mueggler, L. Pesqueira, M. Savva, D. Batra, H. M. Strasdat, R. D. Nardi, M. Goesele, S. Lovegrove, and R. Newcombe, "The Replica Dataset: A Digital Replica of Indoor Spaces," *arXiv preprint arXiv:1906.05797*, 2019.

[23] Y. Zheng, Y. Yang, K. Mo, J. Li, T. Yu, Y. Liu, K. Liu, and L. Guibas, "GIMO: Gaze-Informed Human Motion Prediction in Context," in *European Conference on Computer Vision (ECCV)*, 2022, p. 676–694.