GMMLoc: Structure Consistent Visual Localization With Gaussian Mixture Models

Huaiyang Huang, *Student Member, IEEE*, Haoyang Ye, *Student Member, IEEE*, Yuxiang Sun, *Member, IEEE*, and Ming Liu, *Senior Member, IEEE*

***Abstract*—Incorporating prior structure information into the visual state estimation could generally improve the localization performance. In this letter, we aim to address the paradox between accuracy and efficiency in coupling visual factors with structure constraints. To this end, we present a cross-modality method that tracks a camera in a prior map modelled by the Gaussian Mixture Model (GMM). With the pose estimated by the front-end initially, the local visual observations and map components are associated efficiently, and the visual structure from the triangulation is refined simultaneously. By introducing the hybrid structure factors into the joint optimization, the camera poses are bundle-adjusted with the local visual structure. By evaluating our complete system, namely GMMLoc, on the public dataset, we show how our system can provide a centimeter-level localization accuracy with only trivial computational overhead. In addition, the comparative studies with the state-of-the-art vision-dominant state estimators demonstrate the competitive performance of our method.**

***Index Terms*—Localization, SLAM, visual-based navigation.**

I. Introduction

LOCALIZATION is a crucial capability for robotic navigation, as it can provide the global position and orientation which is essential for high-level applications ranging from path planning to decision-making [1]. Among the available solutions for robot localization, vision-based approaches are becoming increasingly popular due to the widely-used low-cost and light-weight cameras [2], [3]. However, compared to ranging sensors, e.g., LiDARs, the shortcomings of the vision systems are not negligible in that, they generally measure the environment structure in an indirect way and suffer from large appearance variances of the environment [4].

Integrating prior information from scene structures into visual localization systems could alleviate these issues. Along this track, impressive results have been achieved in the recent work [5]–[10]. They usually adopt the pipeline that firstly builds a dense scene structure, and then localizes using visual or visual-inertial sensors with pre-built dense maps [9]. As the structure can be fully reused, this kind of *modality-crossing* formulation between vision and structures allows the localization system to take advantage of both the rich features from visual sensors and the high-precision depth measurements from ranging sensors [10]. However, we observe that there is still a bottleneck on how to efficiently establish the constraints between structure-map elements and local visual measurements. For example, building a kd-tree of a point cloud for searching the correspondences among triangulated visual landmarks takes logarithmic time with respect to the size of the data [11], while downsampling the map would cause a loss of information to a certain extent.

To resolve this issue, in this letter, we present a visual localization system modelling the prior distribution of the visual structure in 3-D space as a mixture of multivariate Gaussian distributions, namely the Gaussian Mixture Model (GMM) [12]. As GMM is a much more compact data quantization method to represent a scene structure compared to the methods based on, for instance, the raw point cloud or voxel grid, the structure information is naturally *concentrated* into this parametric distribution with high-fidelity. Initially tracking the camera pose by an indirect front-end, the proposed method associates the map components with local observations in a flash. The structure constraints are established in a generic form, with which landmarks from the triangulation are well refined. Through optimizing camera poses and landmark positions in the joint Bundle Adjustment (BA), the structure factors are tightly coupled with visual factors from

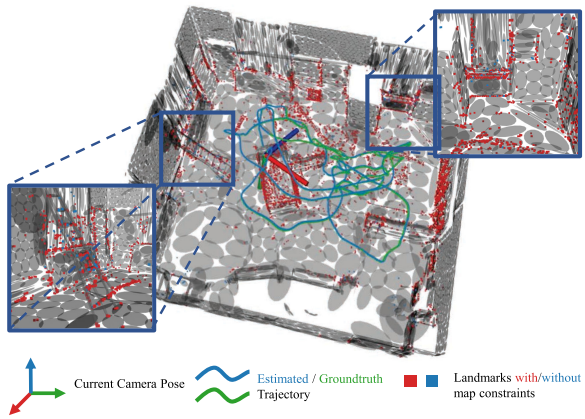


Fig. 1. The proposed method localizes a camera (RGB-Axes) in a prior map represented by GMM. The GMM map is visualized as ellipsoids in $3\boldsymbol{\Sigma}$ bound. As shown in the global and zoom-in view, the estimated visual structure is well aligned with the global map and the trajectory is drift-less compared to the ground truth.

Manuscript received February 24, 2020; accepted June 14, 2020. Date of publication June 25, 2020; date of current version July 11, 2020. This letter was recommended for publication by Associate Editor H. Ryu and Editor Y. Choi upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Project U1713211 and in part by the Research Grant Council of Hong Kong under Project 11210017. *(Corresponding author: Ming Liu.)*

The authors are with RAM-LAB, the Hong Kong University of Science and Technology, Hong Kong 999077, China (e-mail: hhuangat@connect.ust.hk; hy.ye@connect.ust.hk; eeyxsun@ust.hk; liu.ming.prc@gmail.com).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/LRA.2020.3005130

Multiview Geometry (MVG). The experimental results show that the proposed approach achieves an accurate localization performance compared to the state-of-the-art methods, while only a trivial overhead is introduced. A qualitative result is shown in Fig. 1, where the local visual structure is well aligned with the GMM map, and the camera pose in the map frame is accurately recovered. A demo video can be found in our project homepage.[1] We summarize our contributions as follows:

1) Representing the dense structure as GMM, we propose a hybrid structure constraint that ensures the global structure consistency in the visual state estimation.
2) Following a hierarchical scheme, we further propose an efficient method that associates 3-D structure components with 2-D visual observations.
3) Based on the proposed method, we implement GMMLoc, a novel visual localization system that tightly-couples the visual and structural constraints in a unified framework.
4) Comparative experimental results demonstrate the remarkable performance of the proposed system. The additional study on reconstruction accuracy and structure factor supports our claims and confirms the effectiveness of the proposed method.

II. Related Works

A. Visual Localization With Dense Prior Structure

Visual localization is extensively pursued thus an exhaustive review is prohibitive. Here we limit our discussions to the methods which use the dense prior structure. Introducing dense prior structure has been shown to make a significant improvement to both robustness and accuracy in a vision-dominant localization system [5]–[10]. Caselitz *et al.* [5] proposed to associate the landmarks reconstructed from the monocular visual odometry [13] with a point-cloud map. The 7-Degrees of Freedom (DoF) $\mathfrak{sim}(3)$ transform was estimated in an Iterative Closest Point (ICP) scheme. Kim *et al.* [6] formulated the stereo localization as dense direct tracking of the disparity map with the local point-cloud. Ding *et al.* [7] proposed a sliding-window based stereo-inertial localization method with laser-map constraints. They introduced a hybrid optimization method to register the local sparse feature map with the prior laser map. Huang *et al.* [8] modelled the dense structure as an Euclidean Signed Distance Field (ESDF). The visual structure can then be aligned with the implicit surface. Zuo *et al.* [9] proposed MSCKF with prior LiDAR map constraints (MSCKF w/map). A Normal Distribution Transform (NDT)-based method was used to align the stereo reconstruction with the point cloud prebuilt from LiDAR. Ye *et al.* [10] proposed DSL, where surfel constraints were introduced into the direct photometric error. The monocular camera can be localized in a tightly-coupled photometric BA framework. While previous work succeeded in introducing structure constraints, either in a loosely-coupled or a tightly-coupled manner, our method is different in that we quantize the geometry as a GMM, from which we formulate the structure constraint and introduce it into the visual state estimation. The advantages of GMM representation are two-fold: first, it is highly compact, e.g., for the scene in Fig. 1, the whole map consists of only 4500 Gaussian components, of which the data can be stored in an ASCII file of several kilobytes. Therefore it is efficient in both memory and storage; second, such efficiency further accelerates the whole process for the association and establishing of the constraints.

[1][Online]. Available: https://sites.google.com/view/gmmloc/

B. GMMs in State Estimation for Robotics

Early probabilistic registration methods generally interpreted the point cloud data as GMMs by giving each point an isotropic Gaussian variance. This paradigm was first proposed in [11] to overcome the robustness issue of ICP [14] and its variants [15]. Later, Myronenko *et al.* proposed the well-known CPD in [16], where a close-formed solution to the maximization step (M-step) of the EM algorithm was provided. While these methods improved the robustness and accuracy, they are generally slower than ICP-based approaches. To resolve this issue, recently, Gao *et al.* proposed FilterReg [17], where they formulated the expectation step (E-step) as a filtering problem and parameterized the point cloud data as permutohedral lattices. Besides that, in [18], Eckart *et al.* provided an alternative solution by building a multi-scale GMM tree with anisotropic variances and 15–30 fps registration is achieved with the GPU. Similarly, aided by Inertial Measurement Unit (IMU) to recover roll and pitch, Dhawale *et al.* [19] proposed a Monte Carlo localization method with the GMM based on the belief calculation given the depth map of a RGB-D camera. They further showed that GMM can be an efficient environment modelling method for versatile navigation tasks, varying from occupancy analysis [20] to exploration [21].

Motivated by the success of these works, we assume the visual structure is subject to a probabilistic distribution over the Euclidean space, and formulate the constraints in a least-squares manner. We further show how our method works in a vision-dominant localization system other than those based on ranging sensors [15]–[21]. Our method tightly-couples the structure factors with temporal visual factors, and 6-DoF motion parameters are fully recovered without IMU.

III. Method

The flowchart of our system is shown in Fig. 2, where we train the GMM offline from a given dense structure (e.g., point cloud). Every input image is first tracked with a motion-only BA. Then when a keyframe is selected, it will be utilized in the localization module. For every keyframe, we project the GMM map back to the image coordinate. Then local features are associated with map elements, and its corresponding landmark position from triangulation is refined simultaneously. The joint BA optimizes the keyframe poses and the local structure, yielding a drift-less visual localization system. The whole system introduces structural constraints and seamlessly maintains the global consistency.

A. Notations

Throughout the paper, the following notation is used: bold uppercase for the matrices , e.g., $\mathbf{R}$, bold lowercase for the vector e.g., $\mathbf{x}$, and light lower case for the scalar, e.g., θ.

Given a point $\mathbf{x} \in \mathbb{R}^n$ and a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\boldsymbol{\mu} \in \mathbb{R}^n$, $\boldsymbol{\Sigma} \in \mathbb{R}^{n\times n}$, Mahalanobis distance $\|\mathbf{x}-\boldsymbol{\mu}\|_{\boldsymbol{\Sigma}} = (\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$ is used to measure the distance from the point to the distribution. For measuring the distance between two Gaussian components $p(\mathbf{x}|\mathcal{G}_1) = \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, $p(\mathbf{x}|\mathcal{G}_2) = \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ in $\mathbb{R}^n$, we use the Bhattacharyya Coefficient (BC),

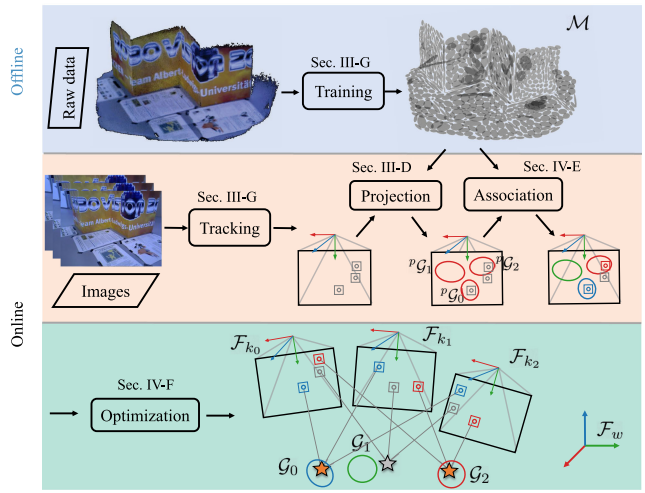


Fig. 2. The flowchart of the proposed system. The model is processed offline. After the visual tracking, the Gaussian components are projected to the image coordinate (shown as red ellipses). In the association step, candidates are first searched in 2-D. Then with the constraints from the map, the correspondence between the local measurement and map component is established (shown in the same color). The back-end optimization is a combination of hybrid constraints (shown in colors) and pure visual constraints (shown in grey).

which is given by:

$$dist_n(\mathcal{G}_1, \mathcal{G}_2) = \frac{1}{8}\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_{\boldsymbol{\Sigma}} + \frac{1}{2}\ln\left(\frac{\det \boldsymbol{\Sigma}}{\sqrt{\det \boldsymbol{\Sigma}_1 \det \boldsymbol{\Sigma}_2}}\right),$$

with $\boldsymbol{\Sigma} = (\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)/2$.

We denote the image collected at the k-th time as I_k and the corresponding frame as $\mathcal{F}_k$. For $\mathcal{F}_k$, the rigid transform $\mathbf{T}_k \in \mathbf{SE}(3)$ maps a 3-D point $\mathbf{x}_i \in \mathbb{R}^3$ in the world frame $\mathcal{F}_w$ to $\mathcal{F}_k$ using ${}^c\mathbf{x}_k = \mathbf{R}_k\mathbf{p}_i + \mathbf{t}_k$, where $\mathbf{T}_k = [\mathbf{R}_k|\mathbf{t}_k]$. $\mathbf{R}_k$ and $\mathbf{t}_k$ are the rotational and translational components of $\mathbf{T}_k$, respectively. Accordingly, ${}^c\mathbf{x}_k$ denotes a 3-D point in $\mathcal{F}_k$. The camera pose, $\mathbf{T}_k$ is parameterized as $\boldsymbol{\xi}_k \in \mathfrak{se}(3)$. We use $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ to denote the projection function: $\mathbf{u} = \pi({}^c\mathbf{x}_k) = \mathbf{K}{}^c\mathbf{x}_k$, where $\mathbf{u}$ is the projected pixel location in the image coordinate. $\mathbf{K}$ stands for the intrinsic matrix.

B. Problem Formulation

The state, measurement and prior are defined as follows:

1) State: $\mathcal{X} = \mathcal{C} \cup \mathcal{L}$, where $\mathcal{C} = \{\boldsymbol{\xi}_0, \boldsymbol{\xi}_1 \dots \boldsymbol{\xi}_m\}$ is the set of keyframe poses in the local covisible map. $\mathcal{L} = \{\mathbf{x}_0, \mathbf{x}_1 \dots \mathbf{x}_n\}$ is the set of all the landmark positions. While $\mathcal{X}$ stands for total states to be optimized, we further denote the set of fixed keyframe poses as $\mathcal{C}' = \{\boldsymbol{\xi}'_0, \boldsymbol{\xi}'_1 \dots \boldsymbol{\xi}'_k\}$, which serves as the prior information in the optimization.

2) Measurement: The measurements consist of 2-D locations of landmarks observed in the pixel coordinate by the different keyframes, denoted as $\mathcal{Z} \doteq \{\mathbf{u}_{ik}\}_{(i,k)\in\mathcal{K}}$. where $\mathbf{u}_{ik}$ is the pixel coordinate of the i-th landmark observed by k-th keyframe and $\mathcal{K}$ is the set of all the visual associations. Similarly, we have $\mathcal{Z}' \doteq \{\mathbf{u}_{ik}\}_{(i,k)\in\mathcal{K}'}$, where $\mathcal{Z}'$ represents the measurements associating prior keyframes with active landmarks.

3) Prior Map: The prior map is denoted as $\mathcal{M} = \{\mathcal{G}_0, \mathcal{G}_1, \dots \mathcal{G}_h\}$, where $\mathcal{G}_j \in \mathcal{M}$ stands for an individual map component (e.g., a voxel in a NDT-based method or a Gaussian distribution for GMMs).

The problem of visual localization against prior map can be formulated as a Maximum A Posteriori (MAP). Instead of using pure visual or visual-inertial information, we introduce the constraints of the pre-built structure, which can be interpreted as defining a prior distribution of the observed visual structure, given as: $p(\mathcal{L}|\mathcal{M})$. This leads the posterior to be factorized as follows:

$$p(\mathcal{X}|\mathcal{Z}, \mathcal{Z}', \mathcal{M}, \mathcal{C}') \propto p(\mathcal{Z}|\mathcal{X}) \cdot p(\mathcal{L}|\mathcal{M}) \cdot p(\mathcal{Z}'|\mathcal{C}', \mathcal{L})p(\mathcal{C})$$
$$= \underbrace{\prod_{i,j} p(\mathbf{u}_{i,j}|\boldsymbol{\xi}_i, \mathbf{x}_j)}_{\text{visual factors}} \cdot \underbrace{\prod_{i,j} p(\mathbf{x}_i|\mathcal{G}_j)}_{\text{structure factors}} \cdot \underbrace{\prod_{i,j} p(\mathbf{u}'_{i,j}|\boldsymbol{\xi}'_i, \mathbf{x}_j) \prod_i p(\boldsymbol{\xi}_i)}_{\text{prior factors}}$$

For the abundant advantages of GMMs as mentioned above, here we model the prior structure as a generic GMM with anisotropic covariances:

$$p(\mathcal{L}|\mathcal{M}) = \sum_{j=0}^{N} w_j p(\mathbf{x}_i|\mathcal{G}_j) = \sum_{j=0}^{N} w_j \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j). \quad (1)$$

In other words, any landmark $\mathbf{x}_i$ should be subject to a prior distribution and its likelihood is given by $p(\mathbf{x}_i|\mathcal{G}_j) \propto \exp(\|\mathbf{x}_i - \boldsymbol{\mu}_j\|_{\boldsymbol{\Sigma}_j})$.

Assuming the noise of measurements is zero-mean Gaussian, maximizing the posterior is equivalent to a least-squares optimization problem, with the objective function as follows:

$$E_{\text{total}} = E_{\text{visual}} + E_{\text{structure}} + E_{\text{prior}}, \quad (2)$$

where different residual terms are defined in the following section.

C. Residual Definitions

1) Visual Factors: Our system follows an indirect formulation of the visual residual, which is also known as the *reprojection error*:

$$\mathbf{e}_{\text{proj}}(\mathbf{x}_i, \boldsymbol{\xi}_k) = \mathbf{u}_{ik} - \pi(\mathbf{R}_k\mathbf{x}_i + \mathbf{t}_k), \quad (3)$$

where $\mathbf{u}_{ik}$ is assumed with a Gaussian noise $\mathcal{N}(0, \boldsymbol{\Sigma}_{ik})$, $\boldsymbol{\Sigma}_{ik} = \sigma_{ik}^2\mathbf{I}_{2\times 2}$ and σ_{ik} is the variance predefined for the local measurement. Given the association set $\mathcal{K}$, visual factor E_{visual} is given as:

$$E_{\text{visual}} = \sum_{(i,k)\in\mathcal{K}} \rho\left(\|\mathbf{e}_{\text{proj}}(\mathbf{x}_i, \boldsymbol{\xi}_k)\|_{\boldsymbol{\Sigma}_{ik}}\right), \quad (4)$$

where $\rho(\cdot)$ is the Huber norm for the robustness in the optimization.

2) Structure Factors: For a landmark $\mathbf{x}_i$ associated with a Gaussian component $\mathcal{G}_j$, the residual term can be derived from the Mahalanobis estimation. Given the likelihood of $\mathbf{x}_i$ as $p(\mathbf{x}_i|\mathcal{G}_j)$, maximizing the log-likelihood is equivalent to minimizing the Mahalanobis distance between $\mathbf{x}_i$ and $\mathcal{G}_j$, yielding the residual term:

$$\mathbf{e}_{\text{str}} = \|\mathbf{x}_i - \boldsymbol{\mu}_j\|_{\boldsymbol{\Sigma}_j}. \quad (5)$$

However, to make all the variables bundle-adjusted, calculating the likelihood or establishing constraints over all the components as probabilistic registration methods [16] is not applicable. Additionally, constraining the landmark position with a 3-D component can somehow be "misleading", as it attempts to minimize the distance from the landmark to the

mean. Inspired by [18], where the authors propose to decompose the anisotropic covariance for accelerating the Mahalanobis estimation, we introduce a hybrid objective function based on the degeneration of different components. As a real-world scene structure is generally constructed by planars, we observe that in a GMM with anisotropic variance directly fitted from a dense point cloud, many components tend to degenerate. Therefore, we detect the degeneration of 3-D Gaussian components in the preprocessing step.

Decomposing the covariance via SVD gives $\mathbf{\Sigma}_j = \mathbf{USV}^T$. For $\mathbf{U} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$, we further let $\mathbf{e}'_3 = \mathbf{e}_1 \times \mathbf{e}_2$ just to ensure it meets the right-hand rule as commonly used in our system, and denote $\mathbf{R} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}'_3]$. Due to the orthonormality between different eigenbases $\mathbf{e}_i$, we have $\mathbf{R} \in \mathbf{SO}(3)$. As the covariance matrix $\mathbf{\Sigma}_j$ is symmetric and positive definite, we have $\mathbf{U} \equiv \mathbf{V}$. Accordingly, the factorization of $\mathbf{\Sigma}_j$ is rewritten as:

$$\mathbf{\Sigma}_j = \mathbf{RSR}^T, \mathbf{S} = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3), \mathbf{R} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}'_3], \quad (6)$$

where $\mathbf{S} = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3)$, $\lambda_1 < \lambda_2 < \lambda_3$ is the diagonal matrix of singular values. In a geometric interpretation, $\mathbf{R}$ is equivalent to the rotation part of the transform from the component coordinate to the world coordinate. In addition a singular value λ_i also stands for the scaling factor according to the data distribution along $\mathbf{e}_i$. A small λ_1 indicates $\mathbf{\Sigma}_j$ tends to be degenerated in rank, or in a geometric interpretation, the component is more similar to a planar. We use $\mathbb{1}(\mathcal{G}_j)$ to indicate whether the i-th component is degenerated or not, given by:

$$\mathbb{1}(\mathcal{G}_j) = \begin{cases} 1 & \lambda_1 \ll \lambda_2 < \lambda_3 \\ 0 & \text{otherwise} \end{cases}. \quad (7)$$

For the degenerated case, we consider the residual term as follows:

$$e_{\text{str\_deg}}(\mathbf{x}_i, \mathcal{G}_j) = \|\mathbf{e}_{j1}(\mathbf{x}_i - \boldsymbol{\mu}_j)\|_{\mathbf{\Sigma}_{\text{str}}}, \quad (8)$$

where $\mathbf{\Sigma}_{\text{str}} = \sigma^2_{\text{str}}\mathbf{I}_{3\times3}$, σ_{str} is the pre-defined variance of structure constraints, which can also be interpreted as a coupling factor for balancing the visual and structural constraints. The effect of σ_{str} is further discussed in Section IV-C. Intuitively, this formulation can be explained as point-to-plane distances, which is efficient for computation and provides a more geometrically explainable formulation of the constraint. Denote the association set as $\mathcal{S}$, the total structure objective function is given by:

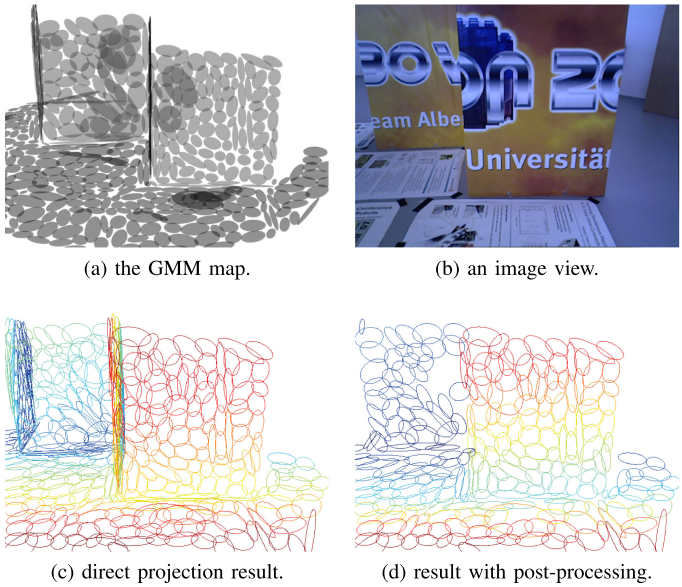
$$E_{\text{structure}} = \sum_{(i,j)\in\mathcal{S}} \left(\mathbb{1}(\mathcal{G}_j) e_{\text{str\_deg}}(\mathbf{x}_i, \mathcal{G}_j) + (1 - \mathbb{1}(\mathcal{G}_j))\, e_{\text{str}}(\mathbf{x}_i, \mathcal{G}_j)\right) \quad (9)$$

3) Prior Factors: In addition to fixed keyframe poses in some visual factors, here we discuss how we deal with the initial estimation. As the proposed method does not aim to solve a global retrieval problem, we consider a prior pose is given at the initialization of the system. In detail, two conditions are discussed:

- if an accurate pose is given (e.g., re-localization from feature map), we set it to the first frame and fix it in the optimization.
- if an initial guess is provided (e.g., from manually assigned), a prior term for constraining the initial pose is added to the optimization, defined as:

$$\mathbf{e}_{\text{init}} = \log(\exp(\boldsymbol{\xi}^{\wedge}_{\text{init}})^{-1} \exp\left(\boldsymbol{\xi}^{\wedge}_{c_0}\right))^{\vee}, \quad (10)$$

where $\boldsymbol{\xi}_{\text{init}}$ and $\boldsymbol{\xi}_{c_0}$ are the preset initial guess and the actual pose for the initial keyframe.



(a) the GMM map. (b) an image view. (c) direct projection result. (d) result with post-processing.

Fig. 3. An example of the GMM projection from the map to the image at the same viewpoint, colored by depth. The filtered result generally has fewer mis-projected components.

D. Projection of the GMM Map

When a keyframe is created, we assume its pose $\boldsymbol{\xi}_k$ is tracked and local observations $\mathcal{O}_k \doteq \{\mathbf{u}_{ik}\}_{i=1\ldots n}$ are detected. To associate the map elements with local observations, we first project the Gaussian components to the image coordinates. With the camera pose $\boldsymbol{\xi}_k$ recovered by the tracking frontend, this projection process can be regarded as a nonlinear transformation of the Gaussian components. Similar discussion can be found in [19], [22]. As the transformation of a point in 3-D Euclidean space is a linear operation, for an individual component $\mathcal{G}_j$, $p(\mathbf{x}_i|\mathcal{G}_j) = \mathcal{N}(\boldsymbol{\mu}_j, \mathbf{\Sigma}_j)$, the density function under $\mathcal{F}_k$ is:

$$p({}^{c_k}\mathbf{x}_i|{}^{c}\mathcal{G}_j) = \mathcal{N}(\mathbf{R}_k\boldsymbol{\mu}_j + \mathbf{t}_k, \mathbf{R}_k\mathbf{\Sigma}_j\mathbf{R}_k^T), \quad (11)$$

while the projection function $\pi(\cdot)$ is nonlinear due to the implicit normalization of the point under image coordinate. A first-order approximation gives:

$$p(\mathbf{u}|{}^{p}\mathcal{G}_j) = \mathcal{N}\left(\pi\left({}^{c}\boldsymbol{\mu}_j\right), \mathbf{J}_\pi|_{{}^{c}\boldsymbol{\mu}_j}\, \mathbf{R}_k\mathbf{\Sigma}_j\mathbf{R}_k^T\mathbf{J}_\pi^T|_{{}^{c}\boldsymbol{\mu}_j}\right) \quad (12)$$

where we denote a 2-D Gaussian component projected from $\mathcal{G}_j$ in the pixel coordinate as ${}^{p}\mathcal{G}_j$. ${}^{c}\boldsymbol{\mu}_j = \mathbf{R}_k\boldsymbol{\mu}_j + \mathbf{t}_k$ is the transformed mean vector and $\mathbf{J}_\pi$ is the Jacobian of $\pi(\cdot)$ with respect to ${}^{c}\boldsymbol{\mu}_j$. An example projection result is shown in Fig. 3.

As we manually "render" the scene with CPU, to generate a photorealistic projection result, we generally use the following criteria to filter the projected components:

- Check if $\mathcal{G}_j$ lies within the image frustum by $({}^{c}\boldsymbol{\mu}_j)_z > 0$.
- For the degenerated component, the angle between viewing ray and $\mathbf{e}_{j1}$ (degenerated axis) of $\mathcal{G}_j$ is checked. If

$$\frac{\langle \mathbf{t}_k - \boldsymbol{\mu}_j, \mathbf{e}_{j1}\rangle}{\|\mathbf{t}_k - \boldsymbol{\mu}_j\|} < \cos\delta_\theta,$$

the component is not observable by the current frame.

- The $\mathbf{\Sigma}_{2\times 2}$ of ${}^{p}\mathcal{G}_j$ is decomposed, and if its singular value $\lambda_{j1} < \lambda_{j2} \ll \delta_\lambda$, the component is considered less representative and is therefore discarded.
- For the remainder we check the occlusion condition. For ${}^{p}\hat{\mathcal{G}}_j = \arg\min_{{}^{p}\mathcal{G}_j} dist_2({}^{p}\mathcal{G}_i, {}^{p}\mathcal{G}_j)$, if $({}^{c}\boldsymbol{\mu}_i)_z < ({}^{c}\boldsymbol{\mu}_j)_z$, $\mathcal{G}_j$ is considered a background component and is supposed to be occluded by the foreground component.

An example filtered result is shown in Fig. 3. The whole projection procedure for a common scene, e.g., the one shown in Fig. 1, can be efficiently finished in several milliseconds using CPU only.

E. Structure Optimization and Association

The complete method for associate local observations with map elements is shown in Alg. 1. Given a keypoint $\mathbf{u}_{ik}$ that can be successfully triangulated either from temporal or static stereo, we select k-nearest 2-D Gaussian components as association candidates, from the set of current projection results $\mathcal{P}$, where the distance metric defined by $\|\mathbf{u}_{ik} - \boldsymbol{\mu}_j\|_{\mathbf{\Sigma}_j}$. This gives the candidate set $\mathcal{P}_i \doteq \{{}^{p}\mathcal{G}_j\}, |\mathcal{P}_i| = k$ (line 1). We then optimize the newly generated landmark position and find the best-fit component (line 2-9). With the triangulated position $\mathbf{x}_i$, we iterate over $\mathcal{P}_i$, and define a sub-problem for optimizing $\mathbf{x}_i$:

$$\hat{\mathbf{x}}_i = \arg\min_{\mathbf{x}_i} \left(\sum_{k'} \|\mathbf{e}_{\text{proj}}\left(\mathbf{x}_i, \boldsymbol{\xi}_{k'}\right)\|_{\mathbf{\Sigma}_{ik'}} + e_{\text{str}}\left(\mathbf{x}_i, \mathcal{G}_j\right) \right). \tag{13}$$

After the optimization, visual residuals are checked to discard outliers. The threshold for visual factors th is determined by $\mathcal{X}^2$-test, where if the $e_{\text{proj}} > th$, we consider this association invalid (line 4-8). Assuming that a valid association is found, we denote the optimal position with the constraint from ${}^{p}\mathcal{G}_j$ as $\hat{\mathbf{x}}_i^j$. We select the final association ${}^{p}\hat{\mathcal{G}}_j$ leading minimum reprojection error (line 7):

$${}^{p}\hat{\mathcal{G}}_j = \arg\min_{{}^{p}\mathcal{G}_j \in \mathcal{P}_i} \sum_{k'} \|\mathbf{e}_{\text{proj}}\left(\hat{\mathbf{x}}_i^j, \boldsymbol{\xi}_{k'}\right)\|_{\mathbf{\Sigma}_{ik'}}.$$

As our method in Section III-D still can not guarantee the map is projected perfectly, mis-projection of components are inevitable. We further verify the likelihood of $\hat{\mathbf{x}}_i$ and then follow an ICP scheme to re-generate the association if the likelihood is low (line 10-20). The procedure can be decomposed into the following two procedures:

1) Given $\hat{\mathbf{x}}_i$, compute $\log(p(\hat{\mathbf{x}}_i|\mathcal{G}_h))$ for $\mathcal{G}_h \in \mathbf{n}(\hat{\mathcal{G}}_j)$. $\mathbf{n}(\hat{\mathcal{G}}_j)$ stands for the set of $\hat{\mathcal{G}}_j$'s neighbours, with the distance defined as $dist_3(\hat{\mathcal{G}}_j, \mathcal{G}_h)$. Then the component with maximum likelihood is assigned to $\hat{\mathcal{G}}_j$.
2) Recompute (13) to get $\hat{\mathbf{x}}_i$.

We iterate until the likelihood of $\mathbf{x}_i$ given $\hat{\mathcal{G}}_j$ is the largest compared to all its neighbours. In this way, the final association not only minimize the reprojection error, but also maximize the likelihood of the landmark.

F. Joint Optimization

With the map constraints, (2) is minimized to solve both keyframe poses and local structure. Similar to [3], the problem is solved by the Levenberg-Marquardt method, which gives a system as follows:

$$\mathbf{H} = \mathbf{J}^T\mathbf{W}\mathbf{J} + \epsilon\mathbf{I}, \qquad \mathbf{b} = -\mathbf{J}^T\mathbf{W}\mathbf{r}, \tag{14}$$

where $\mathbf{J}$ and $\mathbf{r}$ are the stacked Jacobians and residuals, respectively. $\mathbf{W}$ is the weight matrix from stacking the inverse of the covariance for different residual terms as in Section III-C. During the optimization, we further use $\mathcal{X}^2$-test with 95% confidence to filter outliers and perform another round of optimizations with the outliers discarded.

Algorithm 1: Association and Structure Optimization.

Data: $\mathbf{u}_{ik}, \boldsymbol{\xi}_k, \mathbf{x}_i$.
1 $\mathcal{P}_i$ = `candidatesFromProjections`$(\mathbf{u}_{ik}, \mathcal{P})$
2 ${}^{p}\hat{\mathcal{G}}_j \leftarrow$*null*, $\hat{e}_{\text{proj}} \leftarrow \infty$
3 **for** ${}^{p}\mathcal{G}_j$ *in* $\mathcal{P}_i$ **do**
4 $\quad b_{\text{opt}}, \hat{\mathbf{x}}_i, e_{\text{proj}}$ = `optStructure`$(\mathbf{u}_{ik}, \boldsymbol{\xi}_k, \mathcal{G}_j)$
5 $\quad$ `/* b_opt: flag for convergence. */`
6 $\quad$ **if** $e_{proj} < \min(th, \hat{e}_{proj})$ **then**
7 $\quad\quad {}^{p}\hat{\mathcal{G}}_j \leftarrow {}^{p}\mathcal{G}_j$, $\hat{e}_{\text{proj}} \leftarrow e_{\text{proj}}$
8 $\quad$ **end**
9 **end**
10 **if** ${}^{p}\hat{\mathcal{G}}_j \neq$*null* **then**
11 $\quad$ **do**
12 $\quad\quad {}^{p}\hat{\mathcal{G}}_j \leftarrow \hat{\mathcal{G}}_h$
13 $\quad\quad b_{\text{opt}}, \hat{\mathbf{x}}_i, e_{\text{proj}}$ = `optStructure`$(\mathbf{u}_{ik}, \boldsymbol{\xi}_k, \hat{\mathcal{G}}_j)$ **if** b_{opt} **then**
14 $\quad\quad\quad \mathcal{M}_j$ = `findNeighbours`$({}^{p}\hat{\mathcal{G}}_j)$
15 $\quad\quad\quad \hat{\mathcal{G}}_h = \arg\min_{\mathcal{G}_h \in \mathcal{M}_j} \log p(\hat{\mathbf{x}}_i|\mathcal{G}_h)$
16 $\quad\quad$ **end**
17 $\quad$ **while** $\log p(\hat{\mathbf{x}}_i|\hat{\mathcal{G}}_h) > \log p(\hat{\mathbf{x}}_i|\hat{\mathcal{G}}_j)$;
18 **else**
19 $\quad$ **return** *false,* $\mathbf{x}_i$*, null*
20 **end**
21 **return** *true,* $\hat{\mathbf{x}}_i$*,* $\hat{\mathcal{G}}_j$

G. Other Implementation Details

1) Map Processing: We train the GMM from the raw point cloud. The number of total components varies according to different scenes. When initializing the localization system, we load the offline constructed GMM map, decompose the covariance of all the components and check whether they are degenerated. Neighbourhoods are also defined in this procedure.

2) Visual Tracking: Here we follow ORB-SLAM2 [3], an indirect visual SLAM method for camera tracking. Briefly, the frontend extracts ORB features [24] in the incoming frame and associates them with landmarks observed in the previous frame and map. Then, the current camera pose is recovered in a Perspective-n-Point (PnP) scheme. After the initial tracking, the frontend decides whether to insert a keyframe into the backend mainly based on the current tracking quality. The proposed method above happens right after a keyframe is inserted into the backend.

3) Backend Management: Our localization module maintains a local covisibility map, keeps merging similar landmarks and deleting redundant keyframes. For the details of frontend tracking and backend management, we refer the readers to [3].

TABLE I

LEFT: EVALUATION OF THE LOCALIZATION PERFORMANCE ON THE EUROC MAV DATASET. WE REPORT AVERAGE ABSOLUTE TRAJECTORY ERROR(ATE) (M) [23] FOR 5 RUNS ON EUROC MAV DATASET. THE **BEST** AND SECOND-BEST RESULTS ARE HIGHLIGHTED AND THE metric STRIKED OUT STANDS FOR A PARTIAL TRAJECTORY. FOR HFNET, (·) AFTER ATE IS THE TOTAL NUMBER OF FAILURE FRAMES. RIGHT: **SETTINGS** OF THE DIFFERENT METHODS IN THE EXPERIMENT

Seq.	Ours	DSL	MSCKF w/ map	VINS-Mono	ORB-SLAM2	HFNet
V101	**0.030**	0.035	0.056	0.044	0.033	0.062 (3)
V102	**0.023**	0.034	0.055	0.054	0.047	-
V103	0.047	**0.045**	0.087	0.209	0.199	0.118 (79)
V201	**0.018**	0.026	0.069	0.062	0.040	0.083 (10)
V202	**0.020**	0.023	0.089	0.114	0.065	-
V203	~~0.056~~	**0.103**	0.149	0.149	~~0.242~~	0.117 (153)

Method	Input Sensors	Prior Map
ORB-SLAM2	Stereo	(Not Use)
VINS-Mono	Mono + IMU	(Not Use)
MSCKF (w/ map)	Stereo + IMU	Point Cloud
DSL	Mono	Surfel Map
HFNet	Mono	SfM (sparse)
Ours	Stereo	GMM

IV. EXPERIMENTAL RESULTS

We validate the proposed system on the public EuRoC MAV dataset [25]. It provides sequences of stereo images and IMU data streams in three different indoor scenes, with extrinsic calibration, ground truth trajectories and dense reconstruction for two Vicon room configurations (denoted as V1, V2). The main advantages of this dataset are two-fold: first, six sequences including dense scene structures, which supports both cross-modality localization and reconstruction evaluation; second, the aggressive motions and inconsistent illuminations bringing significant challenges for the visual state estimation.

We first evaluate the general localization performance against several state-of-the-art visual or visual-inertial state estimators. Then, we dive into how introducing structure constraints can improve localization performance through the visual structure evaluation, which is followed by a parameter study. Finally, the timing results are provided to prove the real-time performance. All the experiments are performed using a desktop computer equipped with an Intel i7-8700K CPU and 16 GB RAM.

A. General Localization Performance

We compare our method, GMMLoc, with 5 state-of-the-art visual state estimators: our previous work DSL [10], MSCKF with pre-built map (w/ map) [9], VINS-Mono [26], ORB-SLAM2 [13] and HFNet [27]. Among all the methods, GMMLoc, DSL, and MSCKF (w/ map) are similar in that they explicitly introduce the prior-map constraints into a visual state estimation, which we categorize as *dense structure-based* localization methods. On the contrary, HFNet is one the state-of-the-art *sparse structure-based* localization methods, which follows a *SfM-then-localization* pipeline. We also compare the performance of VINS-Mono and ORB-SLAM2, the state-of-the-art VIO/VSLAM methods, following the evaluation protocol in [9], [10]. We further list the settings of different methods in Tab. I. Additionally, to show how the proposed method can improve the estimation accuracy of pure visual odometry, detecting the loop-closure from the image similarity is disabled in the comparison. To evaluate HFNet, we first reconstruct the sparse feature map from SfM on one sequence and then perform localization on the other two under the same scene configuration. We found that the images captured in two *medium* sequences are of the best quality and thus provide a better SfM model for localization.

The localization results are presented in Table I. Generally, our method achieves accurate estimation results compared to other methods. A failure case occurs on sequence V203, where due to the lack of more than 300 frames of the left camera, the indirect frontend in our system, which is similar to our baseline method ORB-SLAM2, cannot manage to track consistently. Therefore, only the accuracy of a partial trajectory is reported. However, we still observe that our method corrects the tracking drift with structure constraints.

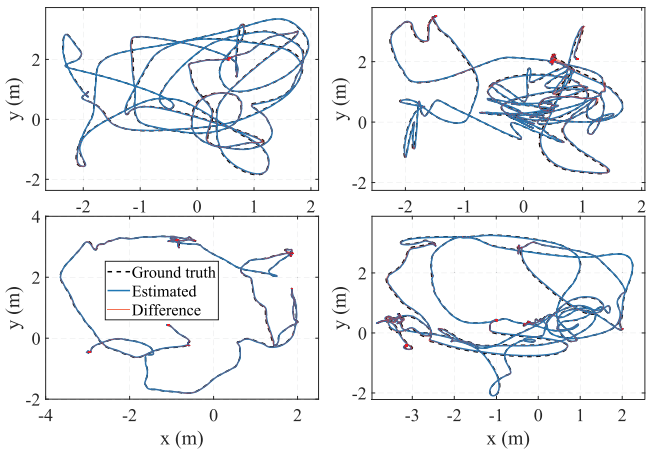


Fig. 4. Qualitative comparison of the estimated trajectory and ground truth on sequences V102 (top left), V103 (top right), V201 (bottom left), V202 (bottom right).

Compared to our previous work DSL, our system achieves a comparable localization accuracy. It is also notable that performance degradation occurs mainly on the difficult sequences. The reasons are two-fold: first, both methods rely on the *projection* procedure for the association, thus the tracking accuracy of the frontend has a significant effect on the association precision; second, our method aims to make a trade-off between accuracy and efficiency, as a consequence of which modelling the scene structure as a GMM does lose some of the structural information. In addition, our method outperforms MSCKF (w/ map) in localization accuracy, while we do not densely reconstruct the scene structure from multiview stereo. Compared to VINS-Mono or ORB-SLAM2, which uses visual-inertial or pure visual information, our method introduces structural constraints and generally improves the localization performance. We also provide some qualitative results in Fig. 4. As shown in the figure, the localization with temporal visual constraints generally has no drift and even the maximum localization error is within an acceptable range (10-20 cm as visualized in the figure).

B. Evaluations of the Local Reconstruction

We evaluate the local structure reconstruction results of our method and ORB-SLAM2 using the ground truth 3-D model

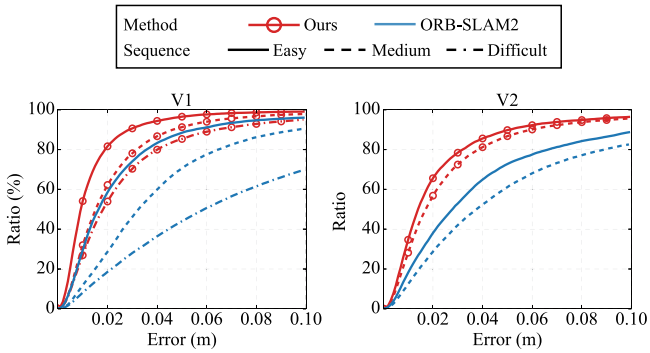


Fig. 5. Evaluation of the visual structure accuracy estimated by Ours and ORB-SLAM2 on different sequences of V1 (left) and V2 (right).

TABLE II
AVERAGE ROOT MEAN SQUARE ERROR (RMSE) (CM) OF THE LOCAL RECONSTRUCTION, LOWER THE BETTER (↓)

Method	v101	v102	v103	v201	v202
Ours	0.098	0.184	2.711	0.408	0.934
ORB-SLAM2	0.338	0.706	6.300	1.620	1.880

provided by the EuRoC Mav dataset. The sparse feature maps generated by the state estimators are aligned and transformed under the map coordinate. The error metric is defined as the RMSE of distances to the nearest neighborhood. A similar evaluation process can be found in [28]. Five sequences on which both methods succeed are selected for the evaluation.

We present the ratio of inliers given an error threshold in Fig. 5 and report the metric results in Table II. As shown in both Fig. 5 and Table II, our method recovers a more accurate local structure, which in turn guarantees the accuracy of local trajectory estimation. Noticeably, as the sequence becomes more challenging, the drawback of pure VO occurs. The estimation drift of VO is not negligible and it can not maintain a globally consistent visual structure. In addition, even under V101 where both methods achieves similar localization performances, our method still outperforms the baseline in terms of structure accuracy. This indicates that introducing the scene structure can also help our system filter the outliers out in two aspects: first, the poses are more accurate in our system, therefore outliers with a larger error in the visual factors, can be more easily detected; second, the BA is constrained by the structure factors, thus the consistency of the visual structure is always maintained.

C. Effectiveness of the Structure Factor

In this section, we further study how the parameters coupling the structure constraints with visual constraints influence the localization performance. For different values of σ_{str}, we perform 5 Monte Carlo runs on each sequence, and report the average ATE with variances in Fig. 6. By increasing σ_{str}, the average localization error approaches to that of the baseline method. This provides an alternative view of the improvement in the localization accuracy compared to pure visual odometry by introducing structure constraints. Especially when $\sigma_{\text{str}} = 1$ m, there is only a trivial improvement to the localization accuracy (even very close to that of the baseline on V201). Note that if we simply discard the structure constraints, the performance should be the same with the baseline. As σ_{str} also represents how

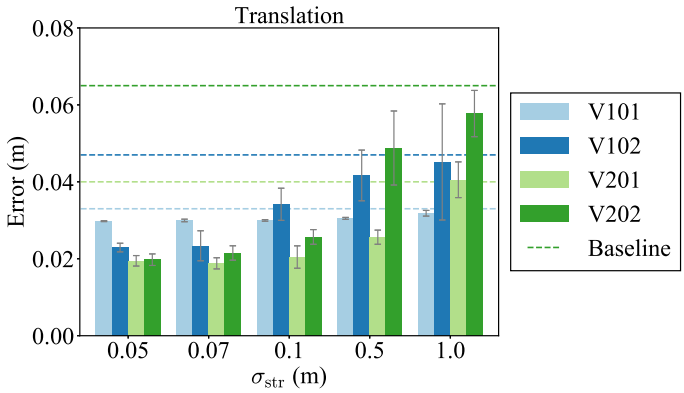


Fig. 6. Parameter study on $\boldsymbol{\sigma}_{\text{str}}$. Localization error (rectangular bar) with variances (grey error bar) w.r.t σ_{str} on 4 sequences are shown. The average ATE of ORB-SLAM2 is shown as dashed lines. Error on different sequences are distinguished by different colors.

TABLE III
TIMINGS FOR THE DIFFERENT MODULES OF GMMLOC, ARE ALL TESTED IN A SINGLE THREAD. THE OVERHEADS COMPARED TO PURE VISUAL ODOMETRY ARE **HIGHLIGHTED**. NOTE THAT THE MAP PREPROCESSING IS PERFORMED ONLY AT THE START OF THE SYSTEM

	Module	Time (ms)
Initialization	**Map Preprocessing**	**15.4 ± 0.0**
Tracking	Feature Extraction	19.1 ± 3.4
	Camera Tracking	15.1 ± 1.6
	KeyFrame Creation	0.1 ± 0.3
	Total	34.5 ± 4.5
Localization	**Map Projection**	**3.5 ± 3.2**
	Initial Association	**0.5 ± 0.3**
	Structure Optimization	**8.2 ± 2.3**
	Local BA	350.5 ± 201.5
	Map Management	52.3 ± 12.6
	Total	409.7 ± 205.5

the optimization weighs structure constraints, we observe that generally, with a low σ_{str} value (in our experiment, 0.05–0.1 m), the estimation can be more consistent over different runs (shown as low variances in Fig. 6).

D. Runtime Analysis

To demonstrate the real-time capability of the proposed method, we report the runtime analysis in Table III. As mentioned in Section I, the projection and association can be rather efficient and the only trivial overhead is introduced to the vision-only backend. Based on the evaluation, it only costs around 10ms, which takes up to around 1/40 in the backend optimization. In addition, as such an overhead only occurs in the backend, the time cost can be even less than 1ms if averaged by the frame rate. This gives solid support for the previous claim that our system is more efficient and has the potential to be applied to embedded platforms.

As reported in MSCKF (w/ map) [9] and DSL [10], MSCKF (w/ map) performs dense reconstruction and NDT-based local registration to localize the camera, which achieves a frequency of around 1.25Hz, while DSL utilizes a modern GPU to render the scene structure for the data association. On the contrary, the proposed method projects the global map elements and associates them with local observations in a flash,

using only a CPU without multi-threading. Yet, our method still exhibits the capability of estimating an accurate trajectory, indicating that it makes a good trade-off between accuracy and efficiency.

V. Conclusion

In this letter, we presented a structure-consistent visual localization method using the GMM as a map representation. Given a camera pose tracked by the indirect front-end, the GMM map is projected back via a non-linear Gaussian transform, and several criteria are applied for a photorealistic projection. Association is performed in three hierarchical steps, searching candidates, finding the component to minimize the reprojection error, and further verify the association with likelihood. In the meantime, the landmark position from triangulation is refined with structural constraints. Finally, the back-end jointly optimizes the visual structure, and the keyframe poses. The experimental results demonstrated the effectiveness of the proposed method. As our method balances accuracy and efficiency well, we believe it has the potential to be applied to onboard platforms in the future.

As the next step, we plan to investigate how to initialize the depth of keypoints from a GMM projection. Additionally, we believe it is also worthy of studying how to introduce some high-level information like semantics to boost the system. Last but not least, introducing IMU factors for a smoother and more robust pose estimation is also promising for increasing the general localization performance.

References

[1] M. Liu and R. Siegwart, "Topological mapping and scene recognition with lightweight color descriptors for an omnidirectional camera," *IEEE Trans. Robot.*, vol. 30, no. 2, pp. 310–324, Apr. 2014.

[2] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 237–267, Feb. 2002.

[3] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[4] G. Pascoe, W. Maddern, and P. Newman, "Direct visual localisation and calibration for road vehicles in changing city environments," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2015, pp. 9–16.

[5] T. Caselitz, B. Steder, M. Ruhnke, and W. Burgard, "Monocular camera localization in 3-D lidar maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 1926–1931.

[6] Y. Kim, J. Jeong, and A. Kim, "Stereo camera localization in 3-D lidar maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–9.

[7] X. Ding, Y. Wang, D. Li, L. Tang, H. Yin, and R. Xiong, "Laser map aided visual inertial localization in changing environment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4794–4801.

[8] H. Huang, Y. Sun, H. Ye, and M. Liu, "Metric monocular localization using signed distance fields," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 1195–1201.

[9] X. Zuo, P. Geneva, Y. Yang, W. Ye, Y. Liu, and G. Huang, "Visual-inertial localization with prior lidar map constraints," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3394–3401, Oct. 2019.

[10] H. Ye, H. Huang, and M. Liu, "Monocular direct sparse localization in a prior 3-D surfel map," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020.

[11] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2-D and 3-D point matching: Pose estimation and correspondence," *Pattern Recognit.*, vol. 31, no. 8, pp. 1019–1031, 1998.

[12] D. A. Reynolds, "Gaussian mixture models," *Encyclopedia Biometrics*, vol. 741, 2009.

[13] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular slam system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[14] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Proc. Sensor Fusion IV: Control Paradigms Data Struct.*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.

[15] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. 3rd Int. Conf. 3-D Digit. Imag. Model.*, 2001, pp. 145–152.

[16] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2262–2275, Dec. 2010.

[17] W. Gao and R. Tedrake, "Filterreg: Robust and efficient probabilistic point-set registration using Gaussian filter and twist parameterization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11095–11104.

[18] B. Eckart, K. Kim, and J. Kautz, "Hgmr: Hierarchical Gaussian mixtures for adaptive 3-D registration," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018.

[19] A. Dhawale, K. Shaurya Shankar, and N. Michael, "Fast Monte-Carlo localization on aerial vehicles using approximate continuous belief representations," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5851–5859.

[20] C. OMeadhra, W. Tabib, and N. Michael, "Variable resolution occupancy mapping using Gaussian mixture models," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 2015–2022, Apr. 2019.

[21] M. Corah, C. OMeadhra, K. Goel, and N. Michael, "Communication-efficient planning and mapping for multi-robot exploration in large environments," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1715–1721, Apr. 2019.

[22] T. D. Barfoot, "State estimation for robotics," 2019.

[23] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D slam systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 573–580.

[24] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to sift or surf," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.

[25] M. Burri *et al.*, "The euroc micro aerial vehicle datasets," *Int. J. Robot. Res.*, 2016. [Online]. Available: http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.abstra ct

[26] T. Qin, P. Li, and S. Shen, "VINS-MONO: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[27] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12716–12725.

[28] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph," *Robot.: Sci. Syst.*, 2015.