

3D Surfel Map-Aided Visual Relocalization with Learned Descriptors

Haoyang Ye¹, Huaiyang Huang¹, Marco Hutter², Timothy Sandy², and Ming Liu¹

Abstract—In this paper, we introduce a method for visual relocalization using the geometric information from a 3D surfel map. A visual database is first built by global indices from the 3D surfel map rendering, which provides associations between image points and 3D surfels. Surfel reprojection constraints are utilized to optimize the keyframe poses and map points in the visual database. A hierarchical camera relocalization algorithm then utilizes the visual database to estimate 6-DoF camera poses. Learned descriptors are further used to improve the performance in challenging cases. We present evaluation under real-world conditions and simulation to show the effectiveness and efficiency of our method, and make the final camera poses consistently well aligned with the 3D environment.

I. INTRODUCTION

Relocalization is one of the essential requirements for autonomous robots. When a navigation system starts or the estimation of the robot pose is lost, the relocalization module is required to provide an initial six-degrees-of-freedom (6-DoF) pose for the system. In many visual localization systems [1]–[3], an initial pose is assumed to be known in achieving the subsequent camera tracking. Thus, how to obtain the initial pose in a 3D environment becomes an important problem.

A GNSS/IMU system is typically used for outdoor applications to provide the initial poses. However, in GNSS-denied environments, relocalization with only on-board sensors is necessary. Cameras are one of the most popular sensors for autonomous robots, thanks to their low cost and light weight, and many visual simultaneous localization and mapping (SLAM) systems [4], [5] take sequential camera data as input to locate the camera as well as to build the map of the surroundings. However, drift is inevitable in visual SLAM systems, causing gaps between the real 3D model and the mapped one and degrading the metric relocalization performance. Structure from motion (SfM) [6] provides more accurate visual structure, and is utilized in many visual (re-)localization-related works [7]–[10]. However, SfM becomes inefficient and time-consuming when the number of input images increases. It is also not flexible to update the unmapped areas of the SfM model with new images.

To efficiently model the 3D structure, 3D lidars are a good choice since they can provide accurate distance mea-

surements and map 3D environments in real time [11]–[13]. Maps built by 3D lidars are usually of higher quality compared to those from visual SLAM systems. Although 3D lidars are expensive and bulky, which may not suit low-cost platforms, leveraging the pre-built 3D lidar map to improve the camera relocalization capability is still promising. One of the most straightforward ways of utilizing the 3D information for relocalization is to directly associate the 3D map lidar points and 2D image features from images by given camera poses. However, in this way, no inter-frame information, e.g., the covisibility of the frames or the trackability of the points in the database, is considered. The covisibility of the frames helps to find more co-observed 3D structures beyond the most similar frame from the database [10], while the trackability of the points helps to keep trackable 3D points among frames and filter unreliable 3D points. This information can help to improve the recall and accuracy of visual relocalization.

In this paper, we propose an efficient method to take the geometric information from the 3D surfel map to build a visual database for 6-DoF camera relocalization. It takes a 3D surfel map, image sequences, and the corresponding camera poses in the 3D map as inputs. The image points are associated with the 3D surfels and form a visual database, which can be further optimized by surfel reprojection constraints. Within the same surfel map, the visual database can be further updated by new image sequences and camera poses. The main contributions of our paper are:

- An efficient way to build and update a visual database using geometric information from a 3D surfel map, which makes the visual database consistent with the 3D geometry.
- Proposed surfel reprojection constraints to further optimize the visual database. By considering surfel information from the 3D model, keyframes and map points are globally optimized, which improves the final relocalization recall and pose accuracy.
- A hierarchical relocalization strategy utilizing the geometric information from the 3D map, which retrieves the global closest frames and utilizes the covisible and map point information from the visual database to estimate 6-DoF camera poses.
- A validation and ablation study in both simulation and real-world experiments to show the effectiveness of the proposed method.

II. RELATED WORK

In this section, we review the work related to visual (re-)localization. That is, given one or few sequential images,

This work was supported by the National Natural Science Foundation of China, under grant No. U1713211, Collaborative Research Fund by Research Grants Council Hong Kong, under Project No. C4063-18G, and HKUST-SJTU Joint Research Collaboration Fund, under project SJTU20EG03, awarded to Prof. Ming Liu.

¹ H. Ye, H. Huang and M. Liu are with RAM-LAB, the Hong Kong University of Science and Technology, Kowloon, Hong Kong.

² M. Hutter and T. Sandy are with Robotic Systems Lab, ETH Zurich, Switzerland.

camera poses in the visual database are estimated. Based on the different types of visual databases, visual (re-)localization can be divided into three categories, 2D image-based, 3D structure-based and hierarchical approaches [10], [14].

2D image-based approaches provide approximate camera poses which correspond to the most similar images in the database. Vector of locally aggregated descriptors (VLAD) [15] and bag of words (BoW) [16] are two commonly used compact representations in 2D image-based approaches. These approaches are most related to solving place recognition or image retrieval problems [17]–[21]. They offer efficient database construction [19], but cannot accurately provide 6-DoF camera poses.

3D structure-based approaches assume a 3D model of the environment is available. The model can be a structure from motion (SfM) model [8], [22]–[25], in which descriptor matching is applied to find associations between 2D image features and 3D map points. Recently proposed scene coordinate regression networks [26], [27] use convolutional neural networks (CNNs) instead of SfM models to present the environments, and provide per-pixel 2D-3D matches. After finding the 2D-3D matches, the camera poses can be estimated by Perspective-n-Point (PnP) solvers [28], [29] inside a RANSAC loop [30]. 3D structure-based approaches provide more accurate 6-DoF camera poses than 2D image-based approaches [14], [19], [31], but updating the SfM model or networks with additional image inputs is not efficient in these approaches since re-building of the SfM model or re-training of the network is required.

Hierarchical approaches combine the advantages of 2D image-based and 3D structure-based approaches. In [7], [9], [32] and [10], the visual relocalization problem is divided into two steps. In the first step, global retrieval based on 2D image-based approaches is applied. It provides a smaller search region for 2D-3D matching for the next step, where camera poses are estimated similar to the 3D structure-based approaches.

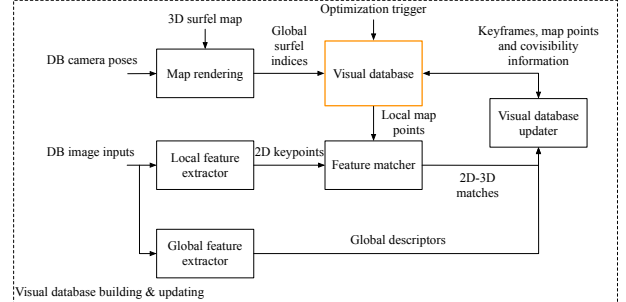
These hierarchical approaches, however, still require a visual database built from SfM [6], [33], which is not flexible to update and typically slow to build. To build the visual database for relocalization efficiently, visual SLAM systems [4], [5] update the visual database continuously since they process the images progressively instead of using all images at once. However, camera pose drift in the SLAM system could cause the visual database to be inconsistent with the true 3D structure.

Based on this observation, our method leverages a 3D surfel map built from 3D lidars, which can be built more efficiently and densely than those from visual-based methods. By utilizing the 3D surfel map, our method makes the visual database consistent with 3D environments. Meanwhile, the visual database can be continuously updated with additional image inputs without re-building the whole database.

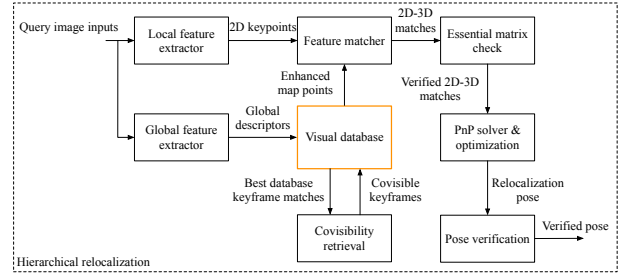
III. METHOD

Our method contains three major steps. The first step (Sec. III-A) is to utilize image inputs, camera trajectory and a

surfel map to build a visual database that is consistent with the 3D environment. Then, surfel reprojection constraints are utilized to optimize the keyframe poses and map point locations (Sec. III-B). Finally, with the optimized visual database, we use hierarchical relocalization to estimate 6-DoF camera poses (Sec. III-C).



(a) Pipeline of visual database building & updating (Sec. III-A and III-B).



(b) Pipeline of visual relocalization (Sec. III-C).

Fig. 1: Pipelines for building visual database and visual relocalization. (a) Sequential image inputs and their corresponding camera poses are utilized to build the visual database leveraging the geometric information from the 3D surfel map. After processing the image inputs, surfel reprojection optimization is triggered to improve the quality of the visual database. (b) When query images arrive, global and local descriptors are extracted for keyframe retrieval and 6-DoF camera pose estimation, respectively.

A. Visual Database Building with a 3D Surfel Map

As shown in Fig. 1a, in the visual database building step, we take image inputs, the corresponding camera poses and a 3D surfel map as inputs. Usually, the camera poses in the 3D environment are not directly available. So there are two practical ways to obtain the camera poses. 1) If an initial guess of the camera pose is available, we can use camera localization algorithms [1]–[3] to calculate the trajectory of the camera in the 3D map. 2) If the 3D map is built by a 3D lidar in motion and rigidly-fixed synchronized camera data is captured at the same time, the camera poses can be estimated by transforming the lidar poses from the lidar mapping algorithms [11], [13] with the extrinsic parameters between the lidar and camera.

When the *visual database building & updating* step starts, the map rendering module loads the surfel map [34] once. We use the same map rendering module as in our previous work [3]. Given a camera pose \mathbf{T}_k^W obtained in one of the above-mentioned two ways, the map rendering module renders a

global surfel indices map in the image plane at timestamp k . Each available pixel in the global surfel indices map is associated with one index of the global surfels in the 3D surfel map. After obtaining the image feature points from the *local feature extractor* module, we can associate 2D feature points $\{\mathbf{p}\}_k$ in image I_k and surfels $\{S\}_k$ in the global surfel indices map, where $\{\cdot\}_k$ denotes a set of points or surfels at timestamp k . 3D map points $\{\mathbf{x}\}_k$ associated with $\{\mathbf{p}\}_k$ are created. The descriptors of $\{\mathbf{x}\}_k$ are set to the corresponding descriptors of $\{\mathbf{p}\}_k$, and the rough positions of $\{\mathbf{x}\}_k$ are set to the center positions of $\{S\}_k$. Besides $\{S\}_k$, we also store neighboring surfel set $\{S\}_k^n$ within the keypoint size of $\{\mathbf{p}\}_k$. $\{S\}_k^n$ is used in Sec. III-C to improve the relocalization recall.

With the next image I_{k+1} and camera pose \mathbf{T}_{k+1}^W , $\{S\}_{k+1}$ can be obtained in the same way. All keyframes, except \mathcal{F}_{k+1} , the keyframe at $k+1$, observing surfel $S \in \{S\}_{k+1}$ are collected as covisible frames $\{\mathcal{F}\}_{k+1}^{co}$. The covisible map points at $k+1$ are defined as $\{\mathbf{x}\}_{k+1}^{co}$, where \mathbf{x} can be observed by \mathcal{F}_α , $\mathcal{F}_\alpha \in \{\mathcal{F}\}_{k+1}^{co}$. We then project $\{\mathbf{x}\}_{k+1}^{co}$ to I_{k+1} and match them with $\{\mathbf{p}\}_{k+1}$. Similar to [4], the matching is performed in the local grid of I_{k+1} and Lowe's ratio test [35] is applied. For each 2D-3D pair, we update the map point's descriptor by finding the best distinctive descriptor among the feature descriptors in $\{\mathcal{F}\}_{k+1}^{co} \cup \mathcal{F}_{k+1}$. For the non-matched 2D image points, we create new 3D map points $\{\mathbf{x}\}_{k+1}$ by the corresponding surfels $\{S\}_{k+1}$.

Since the camera may revisit the same environment during data collection, we want to bound the size of the keyframes and map points in one scene. In our *visual database updater*, if a frame contains 90% of map points which are visible in at least three other keyframes, it is considered as a duplicate frame and culled from the visual database, and if new map points are not visible in the next two keyframes, we will remove them from the visual database. This helps to keep map points likely to be re-observed among keyframes, preserving the best covisibility property.

As we need to retrieve database keyframes sharing the visual content to the query frame, we choose to embed the database keyframes to global descriptors, which are compact to store and efficient to compare.

B. Visual Database Optimization Using Surfel Constraints

In the previous step, the map point positions are assigned to the surfel centers, which may not be accurate enough. The camera trajectory, transformed from lidar poses with varying extrinsic parameters or time offsets, can also introduce uncertainty to the keyframe poses. Since the keyframes and map points are associated through surfels, we can leverage this information to optimize the keyframe poses and map point positions.

Since each map point \mathbf{x}_W in the surfel map's frame \mathcal{F}_W is associated to a surfel S , we can assume the map point is on a global surface with the same normal as the surfel \mathbf{n}_S and the center of surfel \mathbf{x}_S on the surface. The plane coefficient can be written as $\omega_S = [\mathbf{n}_S \ d_S]$, where $d_S = -\mathbf{n}_S^\top \mathbf{x}_S$. \mathbf{x}_W is observed by \mathcal{F}_k and \mathcal{F}_l in their image planes as \mathbf{p}_k and

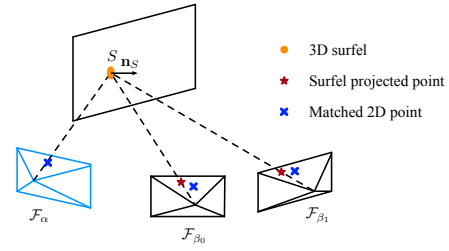


Fig. 2: Reprojection induced by an associated surfel. \mathcal{F}_α is the first keyframe observes the 3D surfel S . Matched 2D points among the keyframes are denoted as blue crosses. The reprojected points in \mathcal{F}_{β_0} and \mathcal{F}_{β_1} induced by the surfel are shown as red stars.

\mathbf{p}_l , respectively. The projection and unprojection functions transform \mathbf{x}_k in \mathcal{F}_k as $\mathbf{p}_k = \Pi(\mathbf{x}_k)$ and $\mathbf{x}_k = \Pi^{-1}(\mathbf{p}_k, \rho_k)$, respectively, where ρ_k is the inverse depth of \mathbf{x}_k in \mathcal{F}_k . Since \mathbf{x}_W is on the plane with coefficient ω_S , it can be written as

$$\mathbf{x}_W^{\rho_k} = \mathbf{R}_k^W \cdot \Pi^{-1}(\mathbf{p}_k, \rho_k) + \mathbf{t}_k^W, \quad \mathbf{n}_S^\top \mathbf{x}_W^{\rho_k} + d_S = 0 \quad (1)$$

where $\mathbf{T}_{(\cdot)}^W$ with $\mathbf{R}_{(\cdot)}^W$ and $\mathbf{t}_{(\cdot)}^W$ as its rotational and translational parts, respectively, transforms points in $\mathcal{F}_{(\cdot)}$ to \mathcal{F}_W . The inverse depth ρ_k can be presented as

$$\rho_k = -\frac{\mathbf{n}_S^\top \mathbf{R}_k^W \cdot \Pi_P^{-1}(\mathbf{p}_k)}{\mathbf{n}_S^\top \mathbf{t}_k^W + d_S}, \quad (2)$$

where $\Pi_P^{-1}(\cdot)$ lifts the image points to the unit image plane. Then, the surfel reprojection error induced by the surfel S between two keyframes at timestamp k and l can be formulated as

$$\mathbf{e}_{sr}(S, k, l) = \Pi((\mathbf{R}_l^W)^{-1} \cdot (\mathbf{x}_W^{\rho_k} - \mathbf{t}_l^W)) - \mathbf{p}_l. \quad (3)$$

Consider all surfels associated with keyframes. The cost function among all surfels can be written as

$$c_{sr} = \sum_j \sum_{\alpha, \beta \in \text{obs}(S_j), \alpha \neq \beta} \|\mathbf{e}_{sr}(S_j, \alpha, \beta)\|^2, \quad (4)$$

where S_j can be observed in \mathcal{F}_α and \mathcal{F}_β . To reduce the redundant constraints, we make \mathcal{F}_α the first frame observing S_j . This reprojection among keyframes induced by a surfel is shown in Fig. 2. Since in Equation 3, \mathbf{x}_W can be presented by \mathbf{T}_k^W , \mathbf{p}_k and ω_S , the variables to be optimized in the final cost function, Equation 4, remain the keyframe poses $\mathbf{T}_{(\cdot)}^W$ only, which can be efficiently solved. After obtaining the optimized keyframe poses, we can further update the map point position by Equation 1 with the first observed keyframe pose.

C. 3D Geometry-Aided Visual Relocalization

As shown in Fig. 1b, we adopt a hierarchical relocalization scheme, containing database frame detection and 6-DoF pose estimation, to perform the relocalization efficiently. Firstly, we retrieve database keyframes which have the most overlapping visual content with the current frame. The global feature extractor will extract a global descriptor from the new query image. Keyframes from the visual database are

detected based on the similarity of the global descriptor of the query image and those of database frames. Our database returns the top- K matched keyframes, which are clustered by the covisibility information, i.e. whether the matched keyframes share the same map points. This procedure returns N_{db} clusters. We then take the keyframe with the highest similarity score in each cluster as the canonical keyframe \mathcal{F}_c . To reduce the final candidate database keyframes and limit the relocalization time, we keep $\min(N_{db}, N_{max})$ canonical keyframes to process the following camera pose estimation.

For each canonical keyframe \mathcal{F}_c , we find N_{co} covisible frames $\{\mathcal{F}\}_{co}$ with highest number of shared map points. The map points visible in $\mathcal{F}_c \cup \{\mathcal{F}\}_{co}$ are matched with 2D feature points in the query frame, providing 2D-3D matches.

Because pose errors exist in the database building step, some map points may not be considered as visible or matchable in $\mathcal{F}_c \cup \{\mathcal{F}\}_{co}$. Only using the map points visible in the local keyframes $\{\mathbf{x}\}_c^F$ might lead to too few good matches, which may cause the PnP solver to be unreliable. Thus, we consider using projected neighboring map points to increase the number of matches. We collect the map points not in $\{\mathbf{x}\}_c^F$ but associated with $\{S\}_c^n$, the neighboring surfels of \mathcal{F}_c , as $\{\mathbf{x}\}_c^N$. Another matching between $\{\mathbf{x}\}_c^N$ and the non-matched 2D feature points is then performed to find more correspondences.

To reduce mismatches, an essential matrix check between \mathcal{F}_c and \mathcal{F}_q is performed before passing the matches to the pose solver. If the matched map points are visible in \mathcal{F}_c , the corresponding keypoint positions are used in \mathcal{F}_c ; if they are not considered as visible, we project the map point positions to \mathcal{F}_c to obtain the keypoint positions. The projected keypoints may be out of \mathcal{F}_c . Instead of throwing away the outside points, we use them in the essential matrix check and keep them if they are considered as inliers after the check. This helps us to retain the map points that are not visible in \mathcal{F}_c but matchable in \mathcal{F}_q .

With the verified 2D-3D matches, we perform PnP with RANSAC [28] to find a rough camera pose for each \mathcal{F}_c containing enough matches. After obtaining the rough pose, we project the local 3D map points to the current image plane again, and find more correspondences in the local grid of the query image. These 2D-3D matches construct a motion-only optimization problem [36] based on reprojection errors, as shown in Equation 5:

$$\begin{aligned} \mathbf{e}_i &= \mathbf{p}_i - \Pi(\mathbf{T}_q^W, \mathbf{x}_{W,i}) \\ c_r &= \sum_i \rho(\mathbf{e}_i^\top \boldsymbol{\Omega}_i^{-1} \mathbf{e}_i) \end{aligned} \quad (5)$$

where \mathbf{T}_q^W is the pose of the query image, Π is the camera projection function, \mathbf{p}_i is the i^{th} keypoint position, $\mathbf{x}_{W,i}$ is the matched map point, ρ is the Huber loss function, and $\boldsymbol{\Omega}_i$ is the covariance matrix from the keypoint scale. Minimizing c_r is a non-linear optimization problem, which can be efficiently solved by Ceres Solver [37].

A pose validation step is used to filter out the wrong poses. We first check the inliers of keypoint-to-map-point (2D-3D) matches provided by the optimized current pose,

n_{in} . If $n_{in} \geq \theta_{in}$, where θ_{in} is a threshold for inliers, the pose is regarded as an inlier pose. Then we check the distance between the current inlier pose and the last inlier pose, d_{in} . If $d_{in} \leq \theta_{dist}$, the current inlier pose is verified and the relocalization system returns a valid final pose for the current query image.

IV. RESULTS

In this section, we evaluate the proposed method with real-world and simulation tests. An ablation study is also conducted to show the effectiveness of our design.

A. Implementation Details

We compare two local feature extractors, ORB [38] and SuperPoint (SP) [39]. For fair comparison with our ORB baseline method, ORB-SLAM [4], we use the local feature detection module from ORB-SLAM, which extracts homogeneously distributed FAST corners [40] among different pyramids of the input images, and the same ORB descriptors [38] for the detected keypoints. A total 1000 ORB keypoints are extracted in each image.

For global feature extractors, we compare two representations of images, the reduced VLAD vector in NetVLAD [17] and bag of binary words (BoBW) [21]. A KD-tree structure of global NetVLAD descriptors is formed for efficient similarity score calculation and the inverted index [21] is used in the BoBW database.

We set the top- $K=30$, $N_{max}=3$ and $N_{co}=5$ to balance the relocalization speed and accuracy. For the pose verification, $\theta_{in}=15$, and $\theta_{dist}=0.3$ m for indoors and $\theta_{dist}=0.8$ m for outdoors are set in all tests, unless specified otherwise.

B. EuRoC Dataset

For the real-world indoor experiments, we use the EuRoC dataset [41], which provides grayscale stereo images captured on a UAV, IMU data, ground-truth poses and a ground-truth lidar point cloud map. In our tests, the ground-truth lidar point cloud map is used to build a surfel map with surfel radius 0.02 m, as described in [3]. For visual database building, the camera poses in the surfel map are estimated by our previous method DSL [3].

We take the monocular images from sequences V102 and V202 to build the visual database of two different rooms V1 and V2, respectively. The sequences V101, V103 are used to test the relocalization performance for room V1; and V201 and V203 are used for the testing of room V2.

We compare two variants with SP or ORB as local feature extractors, ORB-SLAM [4] and NV+SP [10]. NV+SP is the version in [10] taking NetVLAD for global retrieval and SuperPoint for local feature extraction. It requires an offline SfM process to build a 3D model to provide covisibility information for exhaustive inter-frame feature matching. Both ORB-SLAM¹ and our method take sequential image inputs to build the visual database. They both compute the covisibility information on the fly and are more flexible for updating

¹To perform the relocalization with a single image input, ORB-SLAM is set to the relocalization mode without the function of visual odometry.

TABLE I: Evaluation of the relocalization on the EuRoC dataset by different methods. Recall and average ATE (mATE) with unit in [cm] are shown with the format (% / cm). The best and the second-best results are in bold and with underlines, respectively. V102 and V202 are used for building visual databases for the V1 and V2 tests. The proposed method and ORB-SLAM [4] use images sequentially to build the visual database, while NV+SP [10] matches all images to build the visual database by SfM [6].

	Ours (SP)	Ours (ORB)	ORB-SLAM	NV+SP
V101	93.5 / 4.91	<u>94.2</u> / 4.49	55.8 / <u>3.11</u>	99.9 / 2.50
V103	71.5 / 5.25	61.4 / <u>5.95</u>	17.6 / 13.28	93.8 / 20.44
V201	88.3 / 2.82	<u>90.1</u> / 3.41	61.5 / <u>2.96</u>	98.9 / 5.54
V203	<u>53.5</u> / 5.20	26.8 / 10.29	1.0 / <u>9.85</u>	88.7 / 32.39

the visual database. ORB-SLAM estimates keyframe poses and map points up to scale. Hence we align the database keyframe trajectory of ORB-SLAM to the ground camera trajectory according to [42]. Our method uses 3D information from a surfel map, which makes the keyframe poses and map points consistent with the metric environment.

The pose recall at position threshold $\theta_r = 0.3$ m and average absolute trajectories error (mATE) [43] in centimeters of the relocalized poses for the different methods are shown in Table I. Note that NV+SP [10], which uses accurate SfM models [6] to perform localization, has the highest recall in all tests since the covisibility information and feature matching are more exhaustively examined than those from the methods taking sequential images as input. However, it is time-consuming to build and update such SfM models. Our method outperforms ORB-SLAM in terms of recall and has comparative recall with NV+SP in easy sequences V101 and V201. Since the ORB extractor extracts more feature points (~ 1000) than the SuperPoint extractor (~ 500), it gives the system slightly higher recall in easy sequences. However, in difficult sequences, where lighting conditions and view-points are changed greatly, the SuperPoint extractor can provide more consistent feature detection and description than ORB extractor. Thus, the SP variant of our method has a much higher recall. It is worth noting that with pose verification, our method has the lowest mATE in most of the tests. We notice that NV+SP has large pose outliers in difficult sequences V103 and V203. Thus we only show the mATE of NV+SP's poses whose errors are less than 5 m.

To further show the effectiveness of the pose verification introduced in Sec. III-C, we draw the precision-recall curves by changing the numbers of inliers to determine the success of relocalization in Fig. 3. With pose verification, most false relocalization results can be detected and filtered. This leads to higher the precision of the final relocalization results, even in the difficult V103 sequence.

Table II shows an ablation study of the proposed method using different designs of the system. Relocalization recall at $\theta_r = 0.3$ m and mATE in [cm] are shown. For global retrieval, NV denotes NetVLAD [17] and BoBW denotes the BoBW module similar to that in [4], [21] for ORB features. Different points used in our local matching and pose

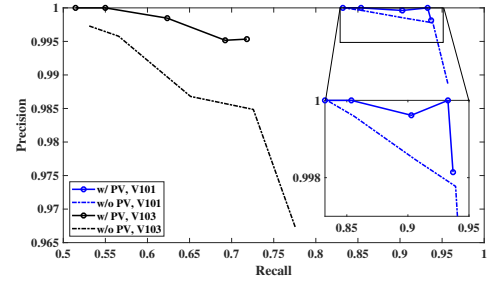


Fig. 3: Precision-recall curves achieved by our method with/without pose verification (PV).

TABLE II: Ablation study on the EuRoC dataset, with VX02 used for building the visual database, and VX01 and VX03 for testing. Relocalization recall and average ATE in [cm], (% / cm), with different local descriptors, SuperPoint (top), ORB (bottom), and other designs of the proposed relocalization approach are shown.

SP				
Global	NV	NV	NV	
Local	FN	F	Naive	
V101	93.5 / 4.91	90.4 / 5.00	86.1 / 6.50	
V103	71.5 / 5.25	65.2 / 5.38	54.0 / 9.87	
V201	88.3 / 2.82	84.7 / 2.90	83.8 / 4.30	
V203	53.5 / 5.20	47.6 / 5.03	45.1 / 6.57	

ORB				
Global	NV	NV	NV	BoBW
Local	FN	F	Naive	FN
V101	94.2 / 4.49	87.0 / 4.60	80.3 / 5.84	88.7 / 4.64
V103	64.1 / 5.95	56.3 / 5.32	42.6 / 6.85	59.8 / 6.95
V201	90.1 / 3.41	85.6 / 3.64	80.4 / 4.15	89.4 / 3.31
V203	26.8 / 10.29	19.8 / 10.97	11.9 / 15.20	18.0 / 8.98

estimation are denoted as local visible points $\{\mathbf{x}\}_c^F$ (F) and neighboring surfel-related map points $\{\mathbf{x}\}_c^N$ (N), as introduced in Sec. III-C. The direct association of the keypoints and the projected 3D surfel centers is denoted as (Naive) with the best matched keyframe. It does not consider the covisibility information nor map point fusion and culling in the visual database. The results show that the global retrieval based on NetVLAD descriptors obtains higher recall than the BoBW-based variant. Since our method involves covisibility information and keeps map points trackable among multiple frames, when performing the 6-DoF camera pose estimation, it has a higher recall and a lower mATE compared to the Naive method. Neighboring surfel-related map points $\{\mathbf{x}\}_c^N$ further improve the relocalization performance by providing more 2D-3D matches.

C. CARLA Simulator

In this section, we evaluate our method in the CARLA simulator [44], which generates image inputs, lidar inputs and ground truth poses. Similar to [3], a 3D surfel map is built by point cloud map from lidar inputs and poses with a 0.1 m average surfel radius.

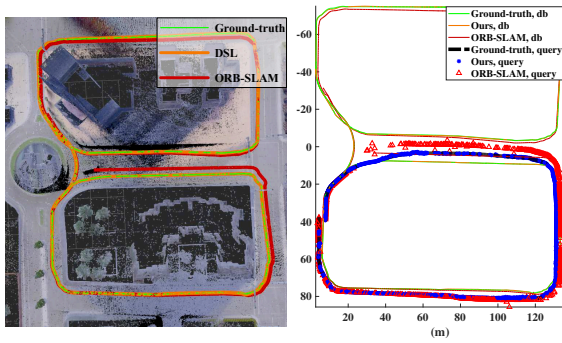
Visual databases are built by different sources of camera poses, as discussed in Sec. III-A: camera poses from a localization algorithm (DSL) [3] and those transformed from

lidar poses with extrinsic parameters between the lidar and camera. We show how the perturbations of camera poses affect the relocalization performance.

In the visual database building step, we add random noises with standard deviation σ to the positions of the camera pose to simulate the perturbations of extrinsic parameters and time offsets between the lidar and camera. After building the visual database, we use the sequence with a car driving in different lanes to test the relocalization. The relocalization recall and average ATE are shown in Table III. From the results, we can see that, with larger pose errors, the performance of relocalization becomes worse. Our optimization introduced in Sec. III-B helps to improve the quality of the visual database. With optimization, the recall increases, and mATE drops, especially for the noisier camera poses.

TABLE III: Relocalization results in CARLA simulator dataset with poses transformed from noisy lidar poses or DSL [3]. The statistics of the corresponding visual databases, relocalization recall and mATE [43] of relocalization are shown. Optimization denotes the visual database optimization introduced in Sec. III-B.

Source of camera poses	Transformed camera poses						DSL poses
	$\sigma = 0.0\text{m}$	$\sigma = 0.2\text{m}$		$\sigma = 0.4\text{m}$			
# of map point	25683	22543		18315			31080
# of keyframes	323	501		559			564
Optimization	N/A	w/o	w/	w/o	w/	w/o	w/
Recall [%]	81.1	78.8	82.0	39.3	74.9	83.7	84.7
mATE [cm]	16.7	49.2	39.2	77.0	52.7	18.4	17.8



(a) Database trajectories

(b) Relocalization results.

Fig. 4: Map building and relocalization results on CARLA simulation. (a) Different trajectories for database building. (b) The trajectory for database building is denoted as “db”; the relocalization results of one query sequence is denoted as “query”. Since our method takes 3D geometric information in the “db” step, metrically correct relocalization results can be provided in the “query” time.

Fig. 4 shows the relocalization results of our method and ORB-SLAM. In our test, ORB-SLAM had scale-drift when building the visual map. Even if we align the camera trajectory [42] to the ground-truth, the drift from the visual system cannot be eliminated completely. This causes large relocalization errors in the regions where the visual system drifts, since the database itself is not well aligned with the

3D environment. Our method takes 3D surfel information in the map building step, and thus can provide metrically correct relocalization results.

D. Runtime

Runtime analysis ² for the different steps of our method are shown in Table IV. The mean process time of the visual database building step is around 50–100 ms, and the relocalization requires less than 50 ms if we extract the local and global descriptors in another independent thread. The visual database optimization runs only once after the sequential images are processed by the visual database updater. Thus, we show the time divided by the number of input images.

We also follow the same procedures as described in [10] to build the visual database for NV+SP. The database building takes more than 20 times the duration of the input sequence, which includes taking 600–900 keyframes for COLMAP [6] to build the SfM model and using the SfM results to find SuperPoint matches and perform triangulation to build a visual database using SuperPoint. Avoiding building an SfM model, our method is more efficient at the time of building the visual database and the visual database can be updated by new sequential images and camera trajectories.

TABLE IV: Runtime of visual database building and relocalization per input image.

Runtime [ms]	EuRoC		CARLA	
	SP	ORB	SP	ORB
Local descriptor				
Feature extraction	16.1	12.7	20.9	14.2
Visual database building	47.7	103.4	85.7	116.6
Map rendering & association	14.6	19.8	20.3	25.0
Visual database updater	31.2	75.3	60.6	88.3
Visual database optimization	1.9	8.3	4.8	3.3
Global descriptor (NV)	25.6		31.9	
Relocalization	30.6	47.8	44.3	47.7
Global retrieval	2.1		2.8	
Local pose estimation	28.5	45.7	41.6	44.9
Image size	752 × 480		800 × 600	

V. CONCLUSION

In this work, we have introduced a camera relocalization algorithm which utilizes a 3D surfel map to build a visual database for relocalization. The proposed method utilizes the surfel map rendering to associate map points with surfels to create covisible information. Surfel constraints among frames are further utilized to improve the quality of the visual database. Given a query image, a hierarchical relocalization is performed with detection of the closest database keyframes and the 6-DoF pose estimation. Both real-world and simulation tests show that our method takes 3D geometric information from the pre-built surfel map to achieve recall comparable with SfM model-based approaches and provide metric relocalization results consistent with the 3D environment. Our future work includes investigating the online updating of the visual database and 3D structure to improve the relocalization ability and handling relocalization in more dynamic environments.

²Run on an Intel i7-8700K CPU with an Nvidia GTX-1080Ti GPU.

REFERENCES

- [1] 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)*. IEEE, 2016, pp. 1926–1931.
- [2] H. Huang, Y. Sun, H. Ye, and M. Liu, "Metric monocular localization using signed distance fields," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.
- [3] H. Ye, H. Huang, and M. Liu, "Monocular direct sparse localization in a prior 3d surfel map," in *2020 International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.
- [4] 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.
- [5] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam," *arXiv preprint arXiv:2007.11898*, 2020.
- [6] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [7] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 2599–2606.
- [8] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua, "Worldwide pose estimation using 3d point clouds," in *European conference on computer vision*. Springer, 2012, pp. 15–29.
- [9] P.-E. Sarlin, F. Debraine, M. Dymczyk, R. Siegwart, and C. Cadena, "Leveraging deep visual descriptors for hierarchical efficient localization," *arXiv preprint arXiv:1809.01019*, 2018.
- [10] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 716–12 725.
- [11] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*, vol. 2, 2014, p. 9.
- [12] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.
- [13] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d lidar inertial odometry and mapping," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.
- [14] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, *et al.*, "Benchmarking 6dof outdoor visual localization in changing conditions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8601–8610.
- [15] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 3304–3311.
- [16] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *null*. IEEE, 2003, p. 1470.
- [17] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.
- [18] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1808–1817.
- [19] T. Sattler, A. Torii, J. Sivic, M. Pollefeys, H. Taira, M. Okutomi, and T. Pajdla, "Are large-scale 3d models really necessary for accurate visual localization?" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1637–1646.
- [20] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.
- [21] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.
- [22] L. Svärm, O. Enqvist, F. Kahl, and M. Oskarsson, "City-scale localization for cameras with known vertical direction," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 7, pp. 1455–1461, 2016.
- [23] L. Svärm, O. Enqvist, M. Oskarsson, and F. Kahl, "Accurate localization and pose estimation for large 3d models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 532–539.
- [24] B. Zeisl, T. Sattler, and M. Pollefeys, "Camera pose voting for large-scale image-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2704–2712.
- [25] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient & effective prioritized matching for large-scale image-based localization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 9, pp. 1744–1756, 2016.
- [26] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "Dsac-differentiable ransac for camera localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6684–6692.
- [27] L. Zhou, Z. Luo, T. Shen, J. Zhang, M. Zhen, Y. Yao, T. Fang, and L. Quan, "Kfnet: Learning temporal camera relocalization using kalman filtering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4919–4928.
- [28] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, p. 155, 2009.
- [29] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *CVPR 2011*. IEEE, 2011, pp. 2969–2976.
- [30] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [31] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe, "Understanding the limitations of cnn-based absolute camera pose regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3302–3312.
- [32] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt, "Scalable 6-dof localization on mobile devices," in *European conference on computer vision*. Springer, 2014, pp. 268–283.
- [33] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixel-wise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.
- [34] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph." *Robotics: Science and Systems*, 2015.
- [35] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [36] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

- [37] S. Agarwal, K. Mierle, and Others, “Ceres solver,” <http://ceres-solver.org>.
- [38] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.
- [39] D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superpoint: Self-supervised interest point detection and description,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 224–236.
- [40] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *European conference on computer vision*. Springer, 2006, pp. 430–443.
- [41] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, “The euroc micro aerial vehicle datasets,” *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.
- [42] S. Umeyama, “Least-squares estimation of transformation parameters between two point patterns,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 4, pp. 376–380, 1991.
- [43] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 573–580.
- [44] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: An open urban driving simulator,” *arXiv preprint arXiv:1711.03938*, 2017.