Low-latency Visual SLAM with Appearance-Enhanced Local Map Building

Yipu Zhao[1], Wenkai Ye[1], and Patricio A. Vela[1]

***Abstract*— A local map module is often implemented in modern VO/VSLAM systems to improve data association and pose estimation. Conventionally, the local map contents are determined by co-visibility. While co-visibility is cheap to establish, it utilizes the relatively-weak temporal prior (i.e. seen before, likely to be seen now), therefore admitting more features into the local map than necessary. This paper describes an enhancement to co-visibility local map building by incorporating a strong appearance prior, which leads to a more compact local map and latency reduction in downstream data association. The appearance prior collected from the current image influences the local map contents: only the map features visually similar to the current measurements are potentially useful for data association. To that end, mapped features are indexed and queried with Multi-index Hashing (MIH). An online hash table selection algorithm is developed to further reduce the query overhead of MIH and the local map size. The proposed appearance-based local map building method is integrated into a state-of-the-art VO/VSLAM system. When evaluated on two public benchmarks, the size of the local map, as well as the latency of real-time pose tracking in VO/VSLAM are significantly reduced. Meanwhile, the VO/VSLAM mean performance is preserved or improves.**

I. Introduction

Augmentation of the feature matching process of VO/VSLAM systems with a local map matching sub-process aids data association and state optimization [1], [2]. Compared with a global map containing all historical 3D points, the local map includes only the subset of 3D points that are hypothesized to be currently visible. Conducting data association and downstream state optimization on a compact local map is more efficient than for the larger global map.

By matching 2D features from the current frame to the local map (which includes 3D points observed at earlier frames), extra long-baseline feature matchings can be extracted and utilized in state optimization; see Figure 1 (top-left) depicting a histogram of matched local map points for ORB-SLAM, where the baseline is measured in terms of how long ago the features were seen (as opposed to how far spatially). These long-baseline matchings contribute to the accuracy and robustness of VO/VSLAM. Not surprisingly, VO/VSLAM systems employing a local map [1], [3] tend to be more accurate and robust than systems relying only on frame-to-frame tracking [4]–[6].

[1]Yipu Zhao, Wenkai Ye, and Patricio A. Vela are with School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA. {yzhao347,wye1206,pvela}@gatech.edu.

This work was supported in part by the China Scholarship Council (CSC Student No: 201606260089) and the National Science Foundation (Award #1816138).

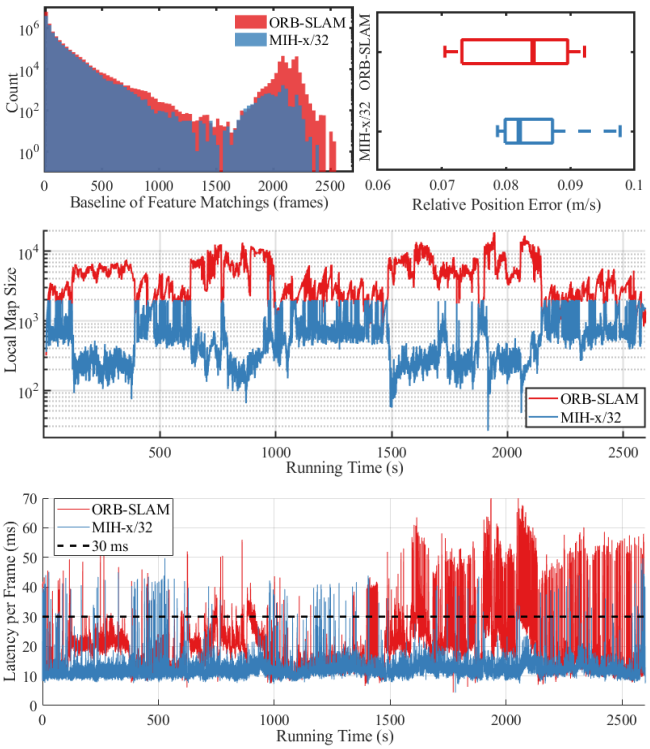

Fig. 1. Latency reduction of the proposed local map building algorithm (MIH-x/32), when integrated into a state-of-the-art VSLAM system (ORB-SLAM [1]). **Top-Left:** Histogram of matched features baselines extracted from local map, with and without proposed algorithm **Top-Right:** Accuracy of VSLAM with or without proposed algorithm, measured with RPE (10-sec window). **Middle:** Size of the local map utilized in VSLAM, with or without proposed algorithm. **Bottom:** Latency profile of real-time pose tracking on the long-term *NewCollege* sequence.

To increase the likelihood of finding and utilizing long-baseline feature matching, it is natural to maintain a history of the 3D points observed earlier in time within the local map. Specific properties or information has been utilized to guide the local map contents to ensure a compact yet relevant of local map, as there is a trade-off between size and search efficiency. The most commonly used property to guide the search of relevant 3D points is *co-visibility*. Co-visibility was introduced for loop closing in VSLAM [7], and later extended to pose tracking [1], [8]–[10]. The assumption of co-visibility being: if an earlier keyframe shares many 3D points with a recent keyframe (i.e. co-visible), then all 3D points observed by the earlier keyframe are likely to be seen also. Co-visibility information is cheap to obtain as the by-product of earlier data association calculations, therefore it can be considered to be an efficient heuristic for local map building. However, co-visibility only utilizes

the relatively-weak temporal prior (i.e. seen before, likely to be seen now). A local map generated with co-visibility could easily grow without bound, and introduce significant latency to VO/VSLAM thereafter. Figure 1 (middle row) includes a plot of the ORB-SLAM local map versus time, where it is seen to occasionally grow to be one to two orders of magnitude more than the number of tracked features per frame (typically on the order of 10^2 to 10^3).

In this work, we propose to enhance the co-visibility local map building step with a strong appearance prior, which will lead to a compact yet relevant local map, a indicated in Figure 1 (middle row) where the proposed local map queried is bounded in size and can be up to an order of magnitude lower that for ORB-SLAM. The idea is straightforward: only those 3D points that are visually similar to currently extracted features are potentially useful in data association (and state optimization thereafter). To utilize the appearance prior efficiently, we propose to index descriptors of historical 3D points with Multi-Index Hashing (MIH) [11]. By querying historical 3D points from a series of hash tables, we can collect the subset of 3D points that are similar to current measurements in appearance/descriptor space. The visually-similar 3D points are then verified with co-visibility, and put together as the local map for the costly computations, e.g. data association and state optimization.

Furthermore, an online table selection algorithm is developed to choose a subset of hash tables that cover the most relevant 3D points. By only querying 3D points from the subset, the overhead on hash table queries is reduced, while the quality of the local map is preserved, as indicated by comparable RPE in Fig 1 (top-right). The table selection process is rooted in the submodular property with regards to the table selection metric (e.g. information gain of feature matchings obtained from each table). Because of the submodular property of table selection metric, a greedy algorithm can achieve near-optimal table selection outcomes with good efficiency properties. Figure 1 (bottom row) shows better bounding of the SLAM latency per frame, with fewer outliers, relative to a 30ms threshold.

The proposed appearance-enhanced local map building method is integrated into a state-of-the-art VO/VSLAM system, ORB-SLAM [1]. When evaluated on multiple public benchmarks, the size of the local map is significantly reduced. More importantly, the proposed method has lower latency than the state-of-the-art VO/VSLAM systems, while remaining one of the best methods in terms of accuracy and robustness. Furthermore, the proposed local map building method is generic; it can be easily extended to other visual(-inertial) SLAM systems utilizing a local map, i.e., [3], [12].

II. Related Works

This section reviews existing works that index 3D points in a map. Two closely-related fields are explored: Vision-based Localization (VBL) & Visual SLAM (VSLAM). Differences between existing works and the proposed work are discussed.

VBL aims to retrieve the 6DoF pose of a visual query (image or video) within a huge, pre-built spatial representation, e.g. a 3D point map. One key component of VBL is to index the spatial representation for efficient query. Co-visibility was introduced to feature-based VBL [13], [14] as a cue to prioritize feature matching efforts. Researchers also proposed alternative indexing methods based on appearance/feature descriptors [15], [16]. Real-valued feature descriptors such as SIFT [17] and SURF [18] are typically indexed offline using a kd-tree. Appearance-based indexing are proven to yield more accurate & robust query results, while co-visibility is more computationally-efficient. Combining both cues was first explored in [19], and further refined in [20], [21]. The work [21] replaced the kd-tree data structure with a faster & more flexible indexing method, inverted multi-index. The appearance-based query results are then filtered with co-visibility. Such a combination scheme is efficient: the VBL system runs real-time on mobile device. Nevertheless, training the inverted index is still an offline process requiring a known 3D map.

Recently, binary feature descriptors such as BRISK [22] and ORB [23] have become popular in VBL since they are more efficient to extract. Conventional indexing data structures like kd-trees are better suited to real-valued descriptors, rather than binary ones, motivating the exploration of alternative indexing methods. For example, [24] proposed to index binary descriptors with randomized trees, which were trained offline from the pre-built 3D map. Hashing has been proven to be a good indexing solution [25], [26] in binary-descriptor VBL. Coarse-to-fine searching schemes are commonly applied in these VBL systems, where an initial hashing query provides the coarse results that are later refined by a linear scan.

Apart from compatibility with binary descriptors, two other properties of hashing make it particularly attractive to online & incremental pose estimation problem, e.g. VSLAM. First, hashing index can be updated efficiently for online processes. It is then possible to generate a more compact and relevant index by updating hash tables, e.g., according to changes in the map & the visibility constraints. Second, hashing relaxes the requirement for database pre-training (or prior offline database generation), therefore enabling VSLAM systems to operate in general and unknown environments. Hashing has been applied to modules of VSLAM where real-time performance is not required. [27] indexed binary descriptors with Locality Sensitive Hashing (LSH) [28], and demonstrated good relocalization performance in a VSLAM system. [29] utilized Multi-Index Hashing (MIH) [11] in the loop closing module of VSLAM.

The proposed work is based on MIH, but with a key enhancement: an online table selection algorithm is developed to reduce the number of hashing queries, therefore enabling MIH to be used in VSLAM modules with real-time requirements, e.g. pose tracking. The local map queried with appearance/feature descriptors is further tailored with a co-visibility check. The final local map is more compact than the ones generated with either co-visibility or appearance only. Running data association and state optimization on the size-reduced local map is more efficient and leads to significant

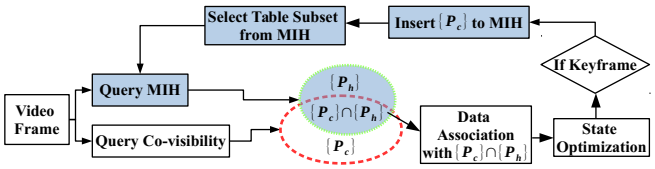

Fig. 2. Framework of the proposed local map building method. The local map built with co-visibility is the red dashed ellipse, while the one built by querying MIH is the green dashed ellipse. Their intersection defines the local map for downstream processing, i.e. data association and state optimization.

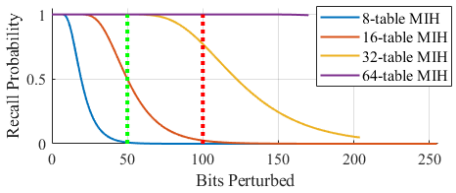

Fig. 3. Simulation results evaluating the recall probability of hashing (the higher the better) vs. the number of bits perturbed for different numbers of tables in the MIH. For 256-bit descriptors, MIH with 32 tables is preferred: it remains high recall even under significant perturbation (50-100 bits).

latency reductions in VSLAM based on a more efficient local map data association step. Furthermore, the quality of the local map (e.g. amount of long-baseline feature matchings) is preserved in the compact local map. Therefore, the performance of VSLAM is preserved. Preliminary quantification of these benefits can be seen in Figure 1 for a single sequence.

III. Local Map Building with Multi-Index Hashing

A diagram of the proposed local map building method is illustrated in Fig 2. The modules of proposed method are highlighted with shaded boxes, while those in a conventional VSLAM pipeline have clear boxes. This section describes the query and insertion stage of MIH. The hash table selection algorithm will be introduced in the next section.

Query MIH. Assume that a frame with m binary descriptors extracted is provided and that the MIH contains t hash tables. Each binary descriptor will trigger a MIH query. In a MIH query, the b-bit binary query descriptor is first separated into t disjoint contiguous substrings. Each substring gets queried with the corresponding hash table for an exact match. Query results from all t hash tables are put together as the final query result. Repeating the MIH query for all binary descriptors from the input frame, aggregate the 3D point set $\{P_h\}$ that satisfy the appearance prior. Its intersecting with the 3D point set $\{P_c\}$ collected with conventional co-visibility builds the final local map $\{P_c\} \cap \{P_c\}$.

Insert to MIH. Updating MIH according to changes in the map & visibility constraints is essential for efficient local map building. As a trade-off between update frequency and computation cost, MIH updates are triggered only for keyframes sent to the mapping thread. Updating MIH in the mapping thread avoids introducing overhead during real-time pose tracking.

For each keyframe, the co-visible 3D points $\{P_c\}$ are inserted into the MIH. Similar to the query process, the b-bit binary descriptor of each 3D point in $\{P_c\}$ is separated into t disjoint contiguous substrings, each of which is of length $\lfloor b/t \rfloor$. Each substring is then inserted into a corresponding hash table. For 3D points already in the hash tables, their entries will be brought to the front of the bucket, making them more likely to be queried in the future.

Choice of hash table number. The amount of hash tables t has strong impact on the performance-efficiency of MIH-based local map indexing. Recall the example of a frame with m features extracted. Each feature will trigger a MIH query consisting of t queries to hash tables. Therefore, the MIH-based local map building has a time complexity of $\mathcal{O}(mt)$, i.e., linear in t. Meanwhile, the space complexity of MIH is $\mathcal{O}(tN2^{\lfloor b/t \rfloor})$, where N is the bucket size in each hash table. The space complexity decreases exponentially with table number t. Therefore, only a certain range of t works in practical applications due to time & space complexity limits.

Apart from time & space complexity, the robustness of local map building against perturbations in binary descriptors is largely decided by hash table number t. Assuming ϵ bits of the query descriptor are perturbed under a uniform distribution, the recall probability (i.e. probability that the query succeeds with a perturbed string) is connected to hash table number t as per [29]:

$$P_{recall}(t, \epsilon) = 1 - t!\,\Theta(\epsilon, t)/t^{\epsilon}, \tag{1}$$

where $\Theta(\epsilon, t)$ is the Stirling partition number [30].

When working with 256-bit binary descriptors such as ORB, the relationship described in Eq 1 is illustrated in Fig 3. The green and red dashed lines indicate example thresholds of bit-wise perturbations in typical SLAM applications. At least 32 tables are needed for high recall probability within the example perturbation levels (vertical dashed lines). Using 64 tables is also possible, but with the drawback of higher overhead due to the linear-growth in time complexity. In the proposed local map indexing method, 32 hash tables are maintained; each table covers an 8-bit descriptor substring.

Choice of bucket size. Another parameter affecting the performance-efficiency of MIH-based local map building is the bucket size N of each hash table. A bucket in MIH is implemented as ring buffer, where only the N most recent 3D points are stored. For the purpose of long-baseline feature matching, it is necessary to keep the entries of 3D points observed earlier in time within the bucket. However, an over-sized bucket will store entries of 3D points that are no longer visible nor relevant. As a consequence, the resulting local map will be less compact and relevant, introducing overhead to data association. In what follows, the bucket size N is set to 10 based on a parameter sweep.

IV. Overhead Reduction with Hash Table Selection

For a frame with m features extracted and a 32-table MIH, the number of hash table queries in local map building is $\mathcal{O}(32m)$. While querying all 32 hash tables provides robustness against severe perturbation, querying a subset of

hash tables is more efficient when the bit-wise perturbation level is low or medium. We propose an online table selection algorithm to identify the minimum subset of hash tables to be queried, which further improve the compactness of local map without performance degeneration.

Formulation. To begin, the metric used for table selection is introduced. Assume F is the full set of *true* feature matchings between current frame and the full local map built with all 32 hash tables. For each hash table T_i, the *true* feature matchings that can be queried from it form a subset $F_i \subset F$, where $\bigcup_{i=1}^{32} F_i = F$. For each hash table T_i, the contribution towards current state optimization can be assessed with the information matrix of subset F_i.

The least squares objective of VO/VSLAM pose tracking is

$$\min \|h(x,p) - z\|^2, \tag{2}$$

where x is the pose of the camera, p are the 3D feature points and z are the corresponding 2D image measurements. The measurement function, $h(x,p)$, is a combination of the $SE(3)$ transformation (world-to-camera) and pin-hole projection. To first-order approximation, the information matrix of the camera pose Ω_x is

$$\Omega_x = \sum H(i)^T \Omega_r(i) H(i) = \sum \Omega_x(i), \tag{3}$$

where $H(i)$ and $\Omega_r(i)$ are the measurement Jacobian and residual information matrix of corresponding *true* matched features. Denote by $\Omega_x(i)$ the pose information matrix derived from a single feature match i.

As introduced for feature subset selection [31], [32], the *logDet* is especially suited for quantifying the contribution of matched features to VO/VSLAM. Therefore, the value of a hash table T_i towards current state optimization can be measured with

$$\log\det(\sum_{i \in F_i} \Omega_x(i)). \tag{4}$$

There is a certain level of overlap between the *true* matched feature subsets for each hash table. In ideal scenario without any perturbation to feature descriptor, the full set of *true* feature matchings can be retrieved from any one of the 32 hash tables, i.e. 100% overlapping between subsets, $\forall i, j \ F_i = F_j = F$. In practice perturbations reduce the subset overlap percentage to less than 100%, and each hash table covers a subset of *true* feature matchings F. Therefore, selecting a subset of hash table is equivalent to a problem of maximum coverage, with the objective formulated as:

$$\max_{S \subseteq \{1,2,\ldots,32\}, |S| \leq k} \log\det(\sum_{i \in \{\bigcup_{h \in S} F_h\}} \Omega_x(i)), \tag{5}$$

where k is the cardinality constraint.

Greedy Solution. The maximum coverage problem is studied in the field of computational theory, where it is known to have submodular properties. Of note,

Theorem 1: [33] Let f be a monotone submoduar function, then greedy algorithms achieve a $(1 - 1/e)$ approximation guarantee to the optimum solution of Eq (5).

Algorithm 1: Online hash table selection algorithm.

Data: feature matching subset from each hash table $\{F_1, F_2, \ldots, F_{32}\}$, cardinality constraint k, target contribution d_{thres}

Result: indices of hash tables selected S

```
foreach feature matching j ∈ ⋃_{i=1}^{32} F_i do
    collect pose information matrix Ω_x(j);
S ← ∅, d_acc = 0;
while |S| < k ∧ d_acc < d_thres do
    foreach i ∉ S do
        d(i) = log det(Σ_{i∈{⋃_{h∈S∪F_i} F_h}} Ω_x(i))
    j ← arg max_i d(i);
    d_acc = d(j);
    S ← S ∪ j;
return S.
```

As proven in [34], *logDet* is submodular & monotone increasing. Solutions to the subset selection problem, and the equivalent hash table selection problem, can be approximated using greedy algorithms. More importantly, a greedy algorithm is guaranteed to be near-optimal, with approximation ratio of $1 - 1/\epsilon$. Based on this outcome, we present a greedy, online hash table selection algorithm in Alg 1. Two control parameters are fixed after parameter sweep: cardinality constraint $k = 8$, target contribution $d_{thres} = 80.0$.

Notice that the above discussion assumes that the *true* feature matchings are known whem performing hash table selection. We assume that the hash table contents are a slowly-varying function of time. Therefore, the hash table subset selection algorithm runs at a lower rate than real-time pose tracking, and only updates the selected subsets at keyframes. Between keyframes, the hash table subset queried for local map building is fixed.

V. Experimental Results

This section evaluates the performance-efficiency trade off of the proposed local map building algorithm on a state-of-the-art VSLAM system, ORB-SLAM [1]. Applying the proposed algorithm to the real-time tracking thread of ORB-SLAM reduces pose tracking latency. Meanwhile, tracking accuracy is either improved (on short sequences) or remains near the same level as canonical ORB-SLAM (on long sequence), and the robustness is preserved (i.e. avoid tracking failure).

Two public benchmarks are used to evaluate the proposed algorithm:

1) *NewCollege* [35], which contains a 43-minutes stereo sequence collected with a robot traversing a campus and adjacent parks. There are multiple loops/revisits within the sequence. The sequence is well-suited for evaluating the long-term performance & efficiency of VSLAM system (with loop closure). Due to the lack of 6DoF pose ground truth, offline Bundle Adjustment is executed with

stereo video, and the jointly optimized camera poses are taken as the ground truth. We only evaluate monocular VSLAM (e.g. with left camera) against the ground truth in this experiment.

2) *EuRoC* [36], which contains 11 stereo-inertial sequences comprising 19 minutes of video, recorded in 3 different indoor environments. Compared with *NewCollege*, videos in *EuRoC* are well-suited for evaluating the short-term performance & efficiency of VO (without loop closure). Ground-truth tracks are provided using motion capture systems (Vicon & Leica MS50). We evaluate only monocular VO implementations on *EuRoC*.

Two performance metrics are used in the experiment. When evaluating the short-term performance of VO on *EuRoC*, *absolute root-mean-square error* (RMSE) between ground truth track and real-time VO estimation is used. When evaluating the long-term performance of VSLAM on *NewCollege*, the Relative Position Error (RPE) [37], [38] is chosen. Compared with absolute RMSE, RPE is less sensitive to the inevitable scale drift of monocular VSLAM. Therefore, it is better for evaluating monocular systems on long-term sequences.

The efficiency of VO/VSLAM is evaluated with the latency of real-time pose tracking per frame, which is defined as the time interval from receiving an image to publishing the pose estimate. Latency of mapping & loop closing is less of a concern in this work due to the relaxed time constraints of those processes.

Performance assessment involves a 10-run repeat for each configuration, i.e., the benchmark sequence, the VO/VSLAM approach and the parameter (number of features tracked per frame). Results for a tested VO/VSLAM configuration are discarded if at least one run experiences track loss. The experiments are conducted on a desktop equipped with an Intel i7 quadcore 4.20GHz CPU (passmark score of 2583 per thread) running the ROS Indigo environment.

A. Online Table Selection vs Fixed Table Subsets

To demonstrate the benefit of online hash table selection (Alg 1), we performed additional 10-run repeats of MIH-based local map building with a predefined set of fixed hash table subsets, ranging 1 table (MIH-1/32) to all 32 tables (MIH-32/32). Results of these tests are compared to MIH-based local map building with online hash table selection, i.e. MIH-x/32 (x = 10).

The latency profiles of different hash table subsets are presented in Fig 4. MIH-x/32 has the lowest latency for data association, when compared to other predefined hash table subsets. The latency of hash table queries is also lower with online hash table selection. Performance evaluation of the methods collected the average RPE (with a 10-sec window), and also logged the average latency of each module in the real-time pose tracking process. Performance (RPE) and efficiency (latency) outcomes are summarized in Fig 5. MIH-x/32 has the lowest latency for pose tracking while preserving the performance of VSLAM relative to the fixed table subsets.

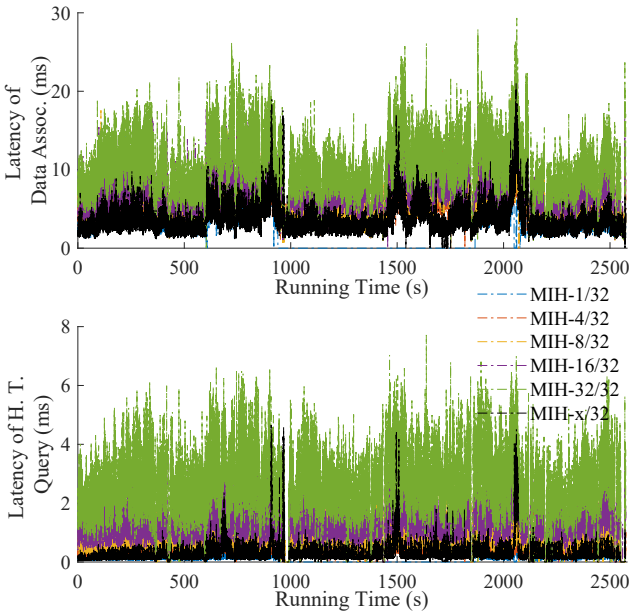

Fig. 4. **Top:** Latency of data association from 1 run on *NewCollege*. **Bottom:** Latency of hash table query (part of data association) from 1 run on *NewCollege*. The first 5 profiles have predefined hash table subsets, e.g. first 1, first 4, etc. The last profile employs online hash table subset selection.

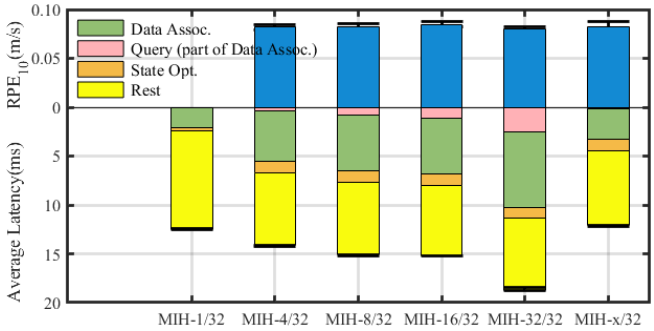

Fig. 5. RPE & latency for different hash table subsets averaged over 10 runs on *NewCollege*. The first 5 columns are the fixed hash table subset methods, e.g. first 1, first 4, etc. The last column employs online selection. No RPE is reported for the single hash table (MIH-1/32) since track loss frequently occurred.

B. Comparison with State-of-the-Art VO/VSLAM

The latency reduction and strong performance of the proposed local map building algorithm is demonstrated by comparing with other state-of-the-art VO/VSLAM systems.

VSLAM. Two state-of-the-art VSLAM systems are chosen as baselines: DSO with loop closure (LDSO) [39] and ORB-SLAM (ORB) [1]. In addition to the proposed MIH-x/32, we integrate two reference methods into ORB-SLAM that enhance co-visibility local map building with simple heuristics. One heuristic is random sampling, i.e. *Rnd*. The other heuristic prioritizes map points with a long track history, denoted as *Long*, since feature points tracked for a long time are more likely to be mapped accurately.

To capture the performance-efficiency trade off of VSLAM systems, we adjust the number of features/patches extracted per frame. All 5 VSLAM systems are configured to run 10-repeats on *NewCollege*, with feature/patch quantities ranging

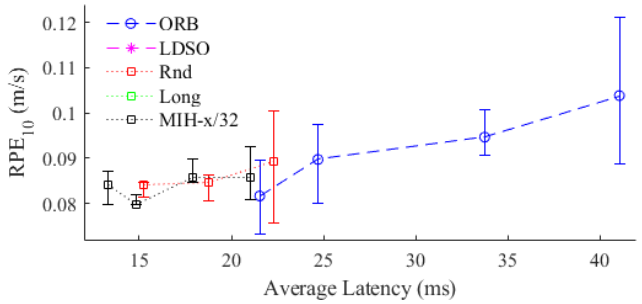

Fig. 6. Latency vs. accuracy on NewCollege monocular sequence. System evaluation involved a sweep of features per frame: 800, 1000, 1500, 2000.

TABLE I

RPE (M/S) AND LATENCY (MS) ON NEWCOLLEGE SEQUENCE

	VSLAM (with loop-closure)			
	Seq.	LDSO	ORB	MIH-x/32
RPE (STD)	RPE_3	-	**0.11** (2e-2)	0.12 **(8e-3)**
	RPE_{10}	-	**0.08** (8e-3)	**0.08 (6e-3)**
	RPE_{30}	-	**0.09 (5e-3)**	0.10 (1e-2)
Latency	Q_1	-	13.2	**10.4**
	Avg.	-	18.3	**12.2**
	Q_3	-	21.5	**13.3**

from 800 to 2000. The RPE under 10-sec window versus the average latency per frame is depicted in Fig 6. Relative to ORB-SLAM, the proposed MIH-x/32 leads to latency reduction for all configurations of feature number. *Rnd* also leads to latency reduction, but not as much as MIH-x/32. The *Rnd* case with 800 features leads to track loss, so it is not plotted. Both LDSO and *Long* failed to track the full *New College* sequence. The accuracy of MIH-x/32 is comparable to the best performing ORB realizations, but with a lower deviation as indicated by the shorter error bars. Lastly, we report the accuracy & latency of the monocular VSLAM systems under the configuration of 800 features per frame in Table I. Three RPE metrics are computed using different sliding windows: 3-sec, 10-sec and 30-sec. In addition to the average RPE over 10-run repeat, the standard deviation (STD) of the RPE is also reported in each cell of Table I. The two heuristics *Rnd* and *Long* are excluded since they both failed to track on the full sequence. The best numbers (lowest average/STD of RPE, lowest latency) are highlighted with bold. The accuracy of MIH-x/32 remains at similar levels as ORB (equal or around 10%), as evaluated on all 3 RPE metrics. More importantly, the latency of proposed method is lower and more consistent than baseline ORB. It is 21%, 33%, and 40% lower for the first quartile, average, and third quartile values.

VO. Two state-of-the-art VO baselines are included: SVO [40] and DSO [41]. For fair comparison, the loop closing module is disabled on all ORB-SLAM variants: canonical ORB, MIH-x/32, *Rnd*, and *Long*. All VO systems are configured to run 10-repeats on *EuRoC* under example configuration (800 features per frame). The short-term performance of VO are evaluated with RMSE, while the efficiency is still assessed via per frame tracking latency. Accuracy & latency results are summarized in Table II. The best value (lowest RMSE, lowest latency) in each row is highlighted with bold in Table II. According to the upper part of Table II, DSO and the 2 local map building heuristics are not robust enough (e.g. frequent track loss). SVO tracks 9 of 11 sequences, but with the highest RMSE over all VO systems. Both ORB baseline and proposed MIH-x/32 track 8 of 11 sequences. Additionally, MIH-x/32 improves the accuracy relative to baseline ORB, with an RMSE average that is 41% lower.

TABLE II

RMSE (M) AND LATENCY (MS) ON EUROC SEQUENCES

	VO (without loop-closure)						
	Seq.	SVO	DSO	ORB	MIH-x/32	Rnd	Long
RMSE	*MH 01 easy*	0.227	0.407	0.027	0.026	**0.025**	-
	MH 02 easy	0.761	-	0.034	**0.031**	0.034	-
	MH 03 med	0.798	0.751	0.041	0.086	**0.035**	-
	MH 04 diff	4.757	-	0.699	**0.293**	0.746	0.329
	MH 05 diff	3.505	-	0.346	**0.197**	-	-
	VR1 01 easy	0.726	0.950	0.057	0.040	**0.034**	-
	VR1 02 med	0.808	**0.536**	-	-	-	-
	VR1 03 diff	-	-	-	-	-	-
	VR2 01 easy	0.277	0.297	0.025	0.032	**0.021**	-
	VR2 02 med	0.722	0.880	0.053	**0.035**	0.216	-
	VR2 03 diff	-	-	-	-	-	-
	Avg.	1.477	0.637	0.160	**0.093**	0.159	0.329
Latency	Q_1	7.4	**5.8**	13.9	11.4	12.0	11.3
	Avg.	**12.6**	16.4	18.4	15.7	16.0	17.7
	Q_3	16.8	19.1	20.7	16.3	**16.1**	21.0

The latency reduction of MIH-x/32 is less significant for these short-term VO sequences, when compared with the previous long-term VSLAM outcomes. Nevertheless, MIH-x/32 has the 2nd lowest average latency among all 6 VO systems, second to SVO. When comparing the 3rd quantile of latency, MIH-x/32 is lower than SVO (by 3%), which suggests that tighter latency bounds can be achieved with the proposed local map building algorithm.

VI. CONCLUSION

This paper demonstrated how an appearance prior can be exploited to build a compact yet relevant local map in VSLAM. Working with the compact local map leads to latency reduction in time-sensitive VSLAM modules, i.e., pose tracking. Meanwhile, the accuracy and robustness of VSLAM is preserved, thanks to the preservation of long-baseline feature associations in the local map. On both long-term VSLAM and short-term VO applications, the proposed algorithm leads to significant latency reduction in real-time pose tracking, while keeping (if not improving) VO/VSLAM performance relative to the baseline variant and having the best performance relative to other state-of-the-art systems.

REFERENCES

[1] 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.

[2] R. Mur-Artal and J. D. Tards, "Visual-inertial monocular SLAM with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[3] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[4] S. Shen, N. Michael, and V. Kumar, "Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft mavs," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 5303–5310.

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

[6] K. Mohta, K. Sun, S. Liu, M. Watterson, B. Pfrommer, J. Svacha, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Experiments in fast, autonomous, gps-denied quadrotor flight," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 7832–7839.

[7] C. Mei, G. Sibley, and P. Newman, "Closing loops without places," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 3738–3744.

[8] H. Strasdat, A. J. Davison, J. M. Montiel, and K. Konolige, "Double window optimisation for constant time visual SLAM," in *IEEE International Conference on Computer Vision*, 2011, pp. 2352–2359.

[9] M. Bürki, I. Gilitschenski, E. Stumm, R. Siegwart, and J. Nieto, "Appearance-based landmark selection for efficient long-term visual localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 4137–4143.

[10] M. A. Nitsche, G. I. Castro, T. Pire, T. Fischer, and P. De Cristóforis, "Constrained-covisibility marginalization for efficient on-board stereo SLAM," in *European Conference on Mobile Robots (ECMR)*. IEEE, 2017, pp. 1–6.

[11] D. Greene, M. Parnas, and F. Yao, "Multi-index hashing for information retrieval," in *35th Annual Symposium on Foundations of Computer Science*. IEEE, 1994, pp. 722–731.

[12] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.

[13] Y. Li, N. Snavely, and D. P. Huttenlocher, "Location recognition using prioritized feature matching," in *European Conference on Computer Vision*. Springer, 2010, pp. 791–804.

[14] S. Choudhary and P. Narayanan, "Visibility probability structure from sfm datasets and applications," in *European Conference on Computer Vision*. Springer, 2012, pp. 130–143.

[15] T. Sattler, B. Leibe, and L. Kobbelt, "Fast image-based localization using direct 2d-to-3d matching," in *International Conference on Computer Vision*, 2011, pp. 667–674.

[16] H. Lim, S. N. Sinha, M. F. Cohen, and M. Uyttendaele, "Real-time image-based 6-dof localization in large-scale environments," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1043–1050.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[18] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European Conference on Computer Vision*. Springer, 2006, pp. 404–417.

[19] T. Sattler, B. Leibe, and L. Kobbelt, "Improving image-based localization by active correspondence search," in *European Conference on Computer Vision*. Springer, 2012, pp. 752–765.

[20] ——, "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, 2017.

[21] S. Lynen, T. Sattler, M. Bosse, J. A. Hesch, M. Pollefeys, and R. Siegwart, "Get out of my lab: Large-scale, real-time visual-inertial localization." in *Robotics: Science and Systems*, 2015.

[22] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *IEEE International Conference on Computer Vision*, 2011, pp. 2548–2555.

[23] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to sift or surf," in *IEEE International Conference on Computer Vision*, 2011, pp. 2564–2571.

[24] Y. Feng, L. Fan, and Y. Wu, "Fast localization in large-scale environments using supervised indexing of binary features," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 343–358, 2016.

[25] J. Cheng, C. Leng, J. Wu, H. Cui, and H. Lu, "Fast and accurate image matching with cascade hashing for 3d reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1–8.

[26] N.-T. Tran, D.-K. Le Tan, A.-D. Doan, T.-T. Do, T.-A. Bui, M. Tan, and N.-M. Cheung, "On-device scalable image-based localization via prioritized cascade search and fast one-many ransac," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 1675–1690, 2019.

[27] J. Straub, S. Hilsenbeck, G. Schroth, R. Huitl, A. Möller, and E. Steinbach, "Fast relocalization for visual odometry using binary features," in *IEEE International Conference on Image Processing*, 2013, pp. 2548–2552.

[28] L. Paulevé, H. Jégou, and L. Amsaleg, "Locality sensitive hashing: A comparison of hash function types and querying mechanisms," *Pattern Recognition Letters*, vol. 31, no. 11, pp. 1348–1358, 2010.

[29] L. Han and L. Fang, "MILD: Multi-index hashing for appearance based loop closure detection," in *IEEE International Conference on Multimedia and Expo*, 2017, pp. 139–144.

[30] R. L. Graham, D. E. Knuth, O. Patashnik, and S. Liu, "Concrete mathematics: a foundation for computer science," *Computers in Physics*, vol. 3, no. 5, pp. 106–107, 1989.

[31] L. Carlone and S. Karaman, "Attention and anticipation in fast visual-inertial navigation," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 1–20, 2019.

[32] Y. Zhao and P. Vela, "Good feature selection for least squares pose optimization in VO/VSLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 3569–3574.

[33] M. X. Goemans and V. Ramakrishnan, "Minimizing submodular functions over families of sets," *Combinatorica*, vol. 15, no. 4, pp. 499–513, 1995.

[34] M. Shamaiah, S. Banerjee, and H. Vikalo, "Greedy sensor selection: Leveraging submodularity," in *IEEE Conference on Decision and Control*, 2010, pp. 2572–2577.

[35] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, "The new college vision and laser data set," *The International Journal of Robotics Research*, vol. 28, no. 5, pp. 595–599, May 2009.

[36] 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.

[37] J. Sturm, W. Burgard, and D. Cremers, "Evaluating egomotion and structure-from-motion approaches using the TUM RGB-D benchmark," in *Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RJS International Conference on Intelligent Robot Systems*, 2012.

[38] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[39] X. Gao, R. Wang, N. Demmel, and D. Cremers, "LDSO: Direct sparse odometry with loop closure," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2198–2204, 2018.

[40] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.

[41] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, 2017.