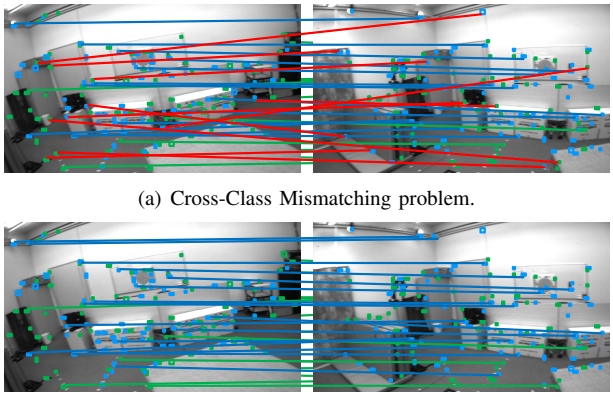
2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

HPF-SLAM: An Efficient Visual SLAM System Leveraging Hybrid Point Features

Xin Su[1], Sebastian Eger[1], Adam Misik[1,3], Dong Yang[1], Rastin Pries[2], and Eckehard Steinbach[1]

***Abstract*— Visual SLAM is an essential tool in diverse applications such as robot perception and extended reality, where feature-based methods are prevalent due to their accuracy and robustness. However, existing methods employ either hand-crafted or solely learnable point features and are thus limited by the feature attributes. In this paper, we propose incorporating hybrid point features efficiently into a single system. By integrating hand-crafted and learnable features, we seek to capitalize on their complementary attributes in both key-point identification and descriptor expressiveness. To this purpose, we design a pre-processing module, which includes extraction, inter-class processing, and post-processing of hybrid point features. We present an efficient matching approach to exclusively perform the data association within the same class of features. Moreover, we design a Hybrid Bag-of-Words (H-BoW) model to deal with hybrid point features in matching and loop-closure-detection. By integrating the proposed framework into a modern feature-based system, we introduce HPF-SLAM. We evaluate the system on EuRoC-MAV and TUM-RGBD benchmarks. The experimental results show that our method consistently surpasses the baseline at comparable speed.**



(a) Cross-Class Mismatching problem.

(b) Efficient processing of hybrid point features (ours).

Fig. 1. The core idea of this work. Firstly, we propose to leverage both hand-crafted (green) and learnable features (blue) within a single system. Secondly, we design an efficient scheme to perform feature matching exclusively within the same feature class, thereby addressing the Cross-Class Mismatching problem in a).

I. Introduction

Visual Simultaneous Localization and Mapping (Visual SLAM) aims to estimate the pose of a mobile agent and simultaneously reconstruct the surrounding environment based on image information. It is a fundamental task in diverse robotic applications, such as autonomous driving, extended reality, and digital twins. Current research in visual SLAM can be divided into two main categories: i) direct methods [1], [2], and ii) feature-based methods [3], [18]. The latter has gained extensive attention due to its well-balanced accuracy and efficiency. Traditional feature-based methods mainly adopt hand-crafted features like SIFT [4] and ORB [5]. These features can be computed efficiently on modern CPUs and exhibit reliable attributes such as rotation- and scale-invariance. However, these methods often struggle in complex scenarios, e.g., varying illumination and viewpoints. Recently, learnable visual features [6], [7], [8] have been applied to robotics perception problems. These have demonstrated competitive performance to hand-crafted features for downstream tasks such as Structure from Motion (SfM) [9] and Visual SLAM [10], [11]. However, learnable features often lack scale information, which limits the performance. In addition, using learnable features with deep descriptors often leads to large maps in terms of storage requirements, which can be burdensome for map management, especially in large-scale environments.

Instead of solely relying on either hand-crafted or learnable features, we propose integrating two classes of features into one visual SLAM system. The purpose is to harness the complementary advantages, which can be distilled into two major aspects: i) complementary distribution of features, and ii) complementary attributes of features. For instance, the ORB features exhibit scale- and rotation-invariance compared to most learnable features such as [6], [7], whereas the latter exhibit better repetitiveness and more distinctive descriptors. To leverage hybrid point features, we introduce an efficient pre-processing module for the extraction, deep-coupled merging, and post-processing of the hybrid point features. This proposed module can be seamlessly integrated into modern visual SLAM systems.

In feature-based visual SLAM systems like ORB-SLAM2, matching feature points is an essential part of data association and plays a fundamental role in the system's overall performance. To match hybrid feature points, a simple approach is assigning an individual matcher for each class of the features, e.g., use SuperGlue [13] to match SuperPoint [6]. However, this approach is slow and leads to redundancy. To ensure efficiency, we propose a single matching scheme to work with the hybrid feature points. An critical issue here is the incorrect matching between feature classes, which we refer

This work was supported by the German Federal Ministry of Education and Research (BMBF) as part of the project "6G-ANNA" with project identification number 16KISK10.

[1]Department of Computer Engineering, School of Computation, Information, and Technology, Munich Institute of Robotics and Machine Intelligence (MIRMI), Chair of Media Technology, Technical University of Munich, Germany, [2]Nokia, Munich, Germany, [3]Siemens Technology, Munich, Germany.

to as Cross-Class Mismatching, depicted in Figure 1 a). Since the data association between features relies on distance-based similarity comparisons (e.g., Hamming distance), incorrect matches can occur if the distance between features from different classes is too low. Notice that their descriptors are computed independently by the respective description algorithms (e.g., Rotated BRIEF and deep descriptor).

Although the cross-class mismatches can be partly eliminated by outlier-rejection mechanisms such as RANSAC, these rejection operations still reduce efficiency. Moreover, the hybrid point features may merge into impure nodes within the Bag-of-Words (BoW) model due to its distance-based nature. The impurity of nodes seriously disrupts the accuracy of the Indirect Index algorithm, which is an essential part of re-localization and loop-closure-detection. An approach to address this issue is to use only one description algorithm for hybrid features in the extraction stage. However, this approach is quasi-hybrid, as it sacrifices the complementary advantages of descriptors. To resolve the cross-class mismatching issue while preserving all benefits of hybrid points features, we present a novel and efficient matching approach by utilizing class descriptors and weighted Hamming distance. The novel matching approach performs matching exclusively within the same feature class. Moreover, we design a Hybrid BoW (H-BoW) model that works properly with hybrid point features.

By integrating the proposed framework into the Collaborative Visual SLAM system [15] (derived from ORB-SLAM2), we build HPF-SLAM, a novel system that can efficiently work with hybrid point features. Compared to ORB-SLAM2 or systems with pure learnable features, our hybrid system demonstrates better robustness in tracking, given its two classes of point features and respective feature pairs. HPF-SLAM produces a denser map with a larger number of robust landmarks (map-points) as it establishes more data associations. By using our designed H-BoW model, the system can effectively employ hybrid point features for Direct- and Indirect-Index algorithms. Notice that the key difference among the methods mentioned above [10], [11] is the class of processed point features. From this perspective, our method can be partly seen as an efficient deep coupling of two systems. Additionally, through the multi-threading architecture in the pre-processing module, our method also ensures real-time performance at the camera frame-rate level. The main contributions of this paper can be summarized as follows:

- We design a real-time pre-processing module to extract and integrate the hybrid point features, and post-process their respective descriptors.
- We present a novel and efficient approach to address the cross-class mismatching problem in data association and introduce a H-BoW model for accelerating the search process, e.g., in feature matching and re-localization.
- We further develop HPF-SLAM that works with hybrid point features. We evaluate the system on EuRoc-MAV and TUM-RGBD datasets. The experimental results show the improved performance of HPF-SLAM over baseline methods while yielding comparable speed.

The rest of the paper is organized as follows. We discuss the related works in Section II and then illustrate the details of our framework in Section III. We present the evaluation in Section IV and conclude our work in Section V.

II. Related Works

A. Feature-Based Visual SLAM

Among feature-based methods, the family of ORB-SLAM systems [18] stands out for its well-balanced accuracy and efficiency. Given the input frame, ORB-SLAM2 begins with the key-point detection and description within a pre-processing module. Then, it divides the complete system into three parallel threads: tracking, local mapping, and loop closing. The multi-threading architecture is one of the major novelties in ORB-SLAM2 and contributes significantly to the overall efficiency. The data association in ORB-SLAM2 is primarily based on the comparison of Hamming distance between the feature descriptors. ORB-SLAM2 also utilizes a BoW model to accelerate the search process in loop-closure-detection, enabling ORB-SLAM2 to efficiently mitigate accumulative errors. The proposed HPF-SLAM is mainly based on ORB-SLAM2.

For the sake of efficiency, Fu et al. [20] proposed omitting the descriptors computation in feature extraction. In their work, the feature matching between successive frames is conducted by a two-stage method: i) estimating key-point correspondences via a motion model and pyramid-based sparse optical flow, and ii) refining correspondences by leveraging the constraints of motion smoothness and epipolar geometry. Moreover, Ferrera et al. [3] pushed the principle of multi-threading into a four-thread architecture (tracking, mapping, state optimization, and loop closing). [3] ensures the forced real-time condition, in other words, no frame loss due to peak processing time. [3] can even be boosted to a frame rate of several hundred Hz while maintaining satisfactory accuracy. In our work, we also implement a multi-threading mechanism in the framework to fulfill the real-time requirement.

B. Deep Learning for Visual SLAM

Learnable methods are intensively researched in the field of visual SLAM. Teed et al. introduced an end-to-end learnable solution, DROID-SLAM [19], which demonstrated significant improvements in accuracy and robustness and showcased the promising prospects of deep learning for future research. Qiu et al. proposed utilizing information of moving objects for dynamic SLAM and introduced Air-DOS [21]. Specifically, they adopted semantic segmentation and human pose detection to model the dynamic elements. They then tracked the dynamic elements by rigid motion constraints and used the dynamic information to rectify the pose estimation.

Learnable feature processing, including extraction and matching, is intensively researched in computer vision. DeTone et al. proposed SuperPoint [6], an efficient self-supervised extractor for joint detection and description of

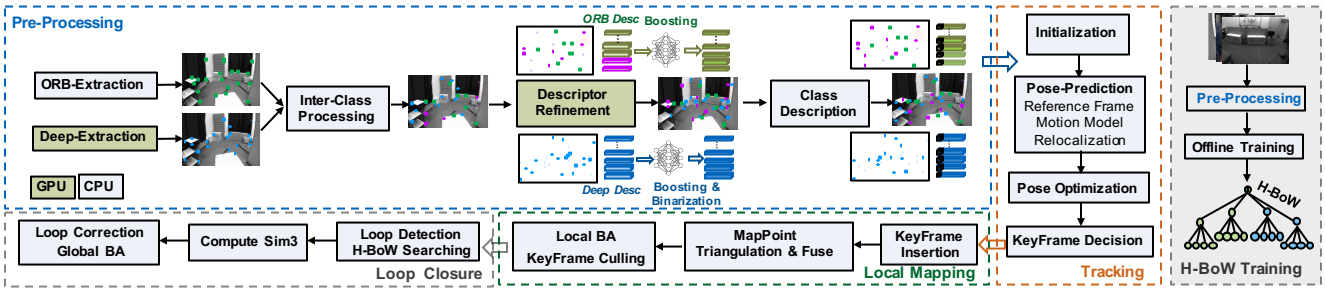


Fig. 2. System overview: given an input frame, we first adopt a multi-threaded pre-processing module to obtain hybrid point features. The subsequent processing follows a structure akin to ORB-SLAM2. To simplify, we list only the key modules and the modules using our proposed matching scheme. We also design an H-BoW model (right) to accelerate the search process (for both Direct- and Indirect-Index algorithms).

key-points. Jérôme et al. presented R2D2 [7] to further learn the repeatability of key-points and the reliability of descriptors. Moreover, Wang et al. introduced FeatureBooster [12], a lightweight neural network to post-process the descriptors to enhance their distinctiveness. It is important to note that our framework is versatile and can be paired with multiple learnable extractors as long as the extractor itself runs in real-time or can be boosted to do so. For feature matching, Sarlin et al. introduced SuperGlue [13], a Graph Neural Network (GNN) architecture with attention mechanism to match SuperPoint features. However, SuperGlue requires intensive computation and is not running in real-time.

For visual SLAM, multiple works proposed modifications to the pre-processing module with learnable feature extractors. Tang et al. introduced GCNv2 [11], a Convolutional Neural Network (CNN)-based extractor with binary descriptors, which can be easily integrated into ORB-SLAM2. Li et al. presented DX-SLAM [10], where they utilized the HF-Net [9] for feature extraction. DX-SLAM also uses the global descriptors obtained from HF-Net for loop-closure-detection and re-localization. However, it is important to note that these methods adopted solely learnable features. These detected features in [11], [10] are processed by the original matching scheme in ORB-SLAM2, which is incapable of simultaneously processing hybrid point features.

C. Hybrid Features for Visual SLAM

Existing methods with hybrid features mainly combine point features with line features [22], or with plane features [23], [24]. For instance, Xu et al. [22] introduced a hybrid visual odometry (VO) approach that combines learnable point features with LSD line features. The matched point-pairs are associated with line features to enhance line matching. Zi et al. [23] combined point features with plane features, where the point features are associated with corresponding planes for refinement. The points and planes are jointly used for pose estimation. Yang et al. [24] further integrated points, lines, and planes in a RGBD-SLAM system. While combining points with lines or planes brings complementary benefits, it often requires additional matching steps and optimization methods due to the inherent differences in feature types. In contrast, this paper proposes to combine points with points, thus enabling efficient matching and optimization.

III. PROPOSED METHOD

In this section, we elaborate on the proposed framework for efficiently processing hybrid point features. Figure 2 gives an overview of the proposed HPF-SLAM. Given the input frames, we use the pre-processing module to detect, integrate, and post-process the hybrid point features. The extracted hybrid point features are fed into an ORB-SLAM2-derived system, where we utilize our proposed matching scheme for data association.

A. Pre-Processing Module

As shown in Figure 2, the pre-processing module consists of four steps:

1) Parallel Feature Extraction.: Taking the gray-scale image as input, we design a two-threading structure to extract hybrid point features efficiently. In the first thread, the ORB feature extraction is performed on a CPU, while in the second thread, we utilize a neural network to jointly detect and describe the learnable features on a GPU. To enhance the feature distribution, we perform Non-Maximum-Suppression (NMS) within each thread. By multi-threading and distributing the operations on CPU and GPU, the extraction step maintains a similar speed as the ORB extractor.

2) Inter-Class Processing.: Despite the NMS in Step 1), the features from different classes can be located too closely, which leads to redundancy. To further improve the feature distribution, we introduce an efficient method to deal with too closely located features, as depicted in Algorithm 1. The core concept is to either perform NMS across the feature classes or merge them into a new ORB-like feature. The criteria here are the distance threshold θ_d and the confidence score θ_s of the learnable features.

3) Descriptor Refinement.: A preliminary pre-condition of dealing with hybrid point features with a single matcher is that the hybrid feature descriptors should have the same shape and format (i.e., binary or float). This requirement is initially not satisfied, as ORB descriptors are binary, while learnable features consist of float values. To address this problem, we propose harmonizing the descriptors format with an efficient post-processing step. We adopt the lightweight FeatureBooster [12] to binarize the deep descriptors (run-time less than 2 ms). By leveraging contextual information of the key-points and an attention mechanism,

Algorithm 1: Inter-Class Processing

Input: ORB Key-points and Descriptors: $\mathbf{K_{orb}}$, $\mathbf{D_{orb}}$
Learnable ones with scores: $\mathbf{K_{lr}}$, $\mathbf{D_{lr}}$, $\mathbf{S_{lr}}$
Threshold for score and distance: θ_s, θ_d
Output: Processed ORB and Learnable Features;
Merged features: $\mathbf{K_{merged}}$, $\mathbf{D_{merged}}$

for $\mathbf{k_{orb}}$, $\mathbf{d_{orb}} \in \mathbf{K_{orb}}, \mathbf{D_{orb}}$ **do**
  for $\mathbf{k_{lr}}$, $\mathbf{d_{lr}}$, $s_{lr} \in \mathbf{K_{lr}}, \mathbf{D_{lr}}, \mathbf{S_{lr}}$ **do**
    $dist = \left\| k_{lr,posi} - k_{orb,posi} \right\|$
    if $dist < \theta_d$ **then**
      if $s_{lr} > \theta_s$ **then**
        delete $\mathbf{k_{orb}}, \mathbf{d_{orb}}$ **from** $\mathbf{K_{orb}}$, $\mathbf{K_{orb}}$
        break;
      else if $s_{lr} < 0.2 * \theta_s$ **then**
        delete $\mathbf{k_{lr}}, \mathbf{d_{lr}}, s_{lr}$ **from** $\mathbf{K_{lr}}$, $\mathbf{D_{lr}}$, $\mathbf{S_{lr}}$
        break;
      else if $dist < 0.2 * \theta_d$ **then**
        $k_{new,posi} = k_{lr,posi} * s_{lr} + k_{orb,posi} * (1\text{-}s_{lr})$
        $k_{new,scale} = k_{orb,scale}$
        delete $\mathbf{k_{lr}}, \mathbf{d_{lr}}, s_{lr}, \mathbf{k_{orb}}, \mathbf{d_{orb}}$,
        insert $\mathbf{k_{new}}$ to $\mathbf{K_{merged}}$,
        break;
      end
    end
  end
end
ComputeORBOrientation($\mathbf{K_{merged}}$)
ComputeORBDescriptor($\mathbf{K_{merged}}$)

FeatureBooster also enhances the distinctiveness of the deep descriptors.

4) Class Description.: To efficiently address the cross-class mismatching problem for hybrid point features, we also describe the class of the feature points. Given the hybrid key-points $\mathbf{k_i}$ with the corresponding descriptors $\mathbf{d_i} \in \mathbb{R}^D$, $i = \{1, \ldots, N\}$, where N is the feature size, we change the first dimension of $\mathbf{d_i}$ (d_i^1) with a binary class-descriptor (1):

$$\mathbf{d_{i,new}} = \begin{cases} \mathbf{d_i} \text{ with } d_i^1 = 0, & \text{for } \mathbf{k_i} \in \text{ORB}; \\ \mathbf{d_i} \text{ with } d_i^1 = 1, & \text{for } \mathbf{k_i} \in \text{Deep}. \end{cases} \tag{1}$$

Modifying the first dimension of d_i (despite the negligible loss in information) aims to maintain the efficiency in subsequent modules. Notice that the descriptor comparison is performed for all kinds of point matching (both key-points and map-points).

We acquire hybrid point features from a single frame by utilizing the pre-processing module. The hybrid point features have two classes of descriptors with class-description. It is worth highlighting that the pre-processing module exhibits versatility, as it can be integrated with diverse real-time learnable feature extractors. Thanks to the binarization of deep descriptors, we can further reduce the storage size of the reconstructed map, which is critical for map management.

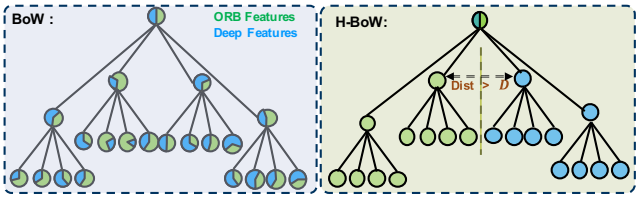


Fig. 3. Instead of mixing visual words with normal BoW models (left), H-BoW (right) comprises two groups of pure nodes in the tree hierarchy, which ensure the accuracy of searching with Indirect-Index algorithms.

B. Data Association with Hybrid Point Features

In this part, we elaborate on the details for matching among hybrid points. Instead of utilizing two independent matchers that are redundant and slow, or adopting one type of descriptors (quasi-hybrid), we introduce a single matcher for two kinds of descriptors. We achieve this by utilizing the class information and weighted Hamming distance.

In systems like ORB-SLAM2, the establishment of feature pairs is principally based on the comparison of descriptor similarity. In this work, we use weighted Hamming distance as the similarity metric to match hybrid binary descriptors. Specifically, the distance C between two descriptors $\mathbf{d_1}$, $\mathbf{d_2} \in \mathbb{R}^D$ is computed with a weight $\mathbf{w} \in \mathbb{R}^D$, as formulated in (2):

$$C(\mathbf{d_1}, \mathbf{d_2}) = \sum_{i=1}^{D} w^i * (d_1^i \oplus d_2^i), \tag{2}$$

where D is the dimensionality of the descriptor, $w^1 = D$ and $w^i = 1$, for $i = \{2, \ldots, D\}$. As mentioned above, d_1^1 and d_2^1 represent the feature classes. Thus, the distance metric (2) meets the condition (3):

$$\begin{cases} C(\mathbf{d_1}, \mathbf{d_2}) \geq D, & \text{if } \mathbf{d_1}, \mathbf{d_2} \text{ in different class;} \\ C(\mathbf{d_1}, \mathbf{d_2}) \leq D - 1, & \text{if } \mathbf{d_1}, \mathbf{d_2} \text{ in same class.} \end{cases} \tag{3}$$

The distance between cross-class features is consistently larger than the maximal distance of same-class features and thus can be easily filtered out. It is worth highlighting that distance comparison is the cornerstone for all kinds of data association with points (2D-2D, 2D-3D, 3D-3D). The utilization of Equation 2 is an efficient and effective solution for the cross-class mismatching problem, as no additional outlier-rejection mechanisms are required.

C. Hybrid Bag-of-Words

Bag-of-Words (BoW) are frequently adopted in visual SLAM to accelerate the search process. To ensure the performance, the nodes of BoW should be appropriately assigned. However, traditional Bag-of-Words (BoW) models encounter the cross-class mismatching problem when dealing with hybrid point features, as their nodes become a mixture of hybrid features during the training process. These impure nodes can disrupt the Direct- and Indirect-Index algorithms, further undermining the performance in feature matching, re-localization, and loop-closure-detection. To solve this problem, we introduce the H-BoW model, depicted in Figure 3.

We utilize the distance metric from Equation 2 in the *K-Means* algorithm while constructing the tree model. This ensures that the cluster centers at each level of the resulting hierarchy are exclusively grouped and updated by visual words from the same class. These nodes inherently carry class information, meeting Condition (3). As a result, the H-BoW covers the hybrid point features with two sets of pure nodes at each level, as in Figure 3. In comparison to traditional BoW models, the nodes in H-BoW ensure purity and maintain distance between different feature classes. The purity of H-BoW nodes plays a critical role in performance. For instance, the efficiency of feature matching (Direct-Index) and the accuracy of re-localization and loop-closure-detection (Indirect-Index).

IV. EXPERIMENTAL EVALUATION

A. Experiment Setup

Datasets We evaluate the proposed HPF-SLAM on two public datasets with different camera types: EuRoc-MAV [25] and TUM-RGBD [26]. The EuRoc-MAV dataset is recorded in a large ETH machine hall and two small rooms. It is collected by a Micro Aerial Vehicle (MAV) equipped with stereo cameras. EuRoc-MAV dataset contains 11 sequences of different difficulty levels, annotated with accurate trajectory ground truth. The TUM-RGBD is collected in multiple scenarios with an RGBD camera. In the experiment, we compare seven representative sequences from three scenarios.

Implementation Details We adopt the officially pretrained SuperPoint midel [6] as the learnable feature extractor. Note that we do not perform any fine-tuning for the sake of out-of-domain ability. We conduct the quantitative experiments on a setup with an Intel Core i9-13900K CPU, 64 GB RAM, and a commercial-level NVIDIA RTX 4070 GPU. The proposed framework is implemented in C++, on Ubuntu 20.04 LTS, with supporting dependencies such as OpenCV, Eigen, g2o, ROS, and LibTorch.

B. Trajectory Evaluation

Like other methods, we measure tracking accuracy using the Root Mean Square of Absolute Trajectory Error (RMS-ATE). For each sequence in the two datasets, we conduct five executions and report the median result. In the ORB SLAM2 baseline experiment, we employ the system from [15] with 1200 ORB features. Additionally, the proposed framework also offers a working mode utilizing solely learnable features. For comparison and ablation study, we also run the framework with pure SuperPoint, denoted as HPF-SLAM (SuperPoint), with a setup of 1000 features.

Results on EuRoc-MAV We compare the proposed framework with other state-of-the-art methods, including VINS-Mono [27], OV^2-SLAM [3] (real-time condition), and learnable methods such as DF-SLAM [28], Attention-SLAM [29] and LIFT [30]. The results are presented in Table I, where "-" indicates unavailable value in literature and "X" indicates tracking failures. We omit the experiments with sequence V203, as most of the methods fail in this case. We highlight the best performance among sequences in bold. The

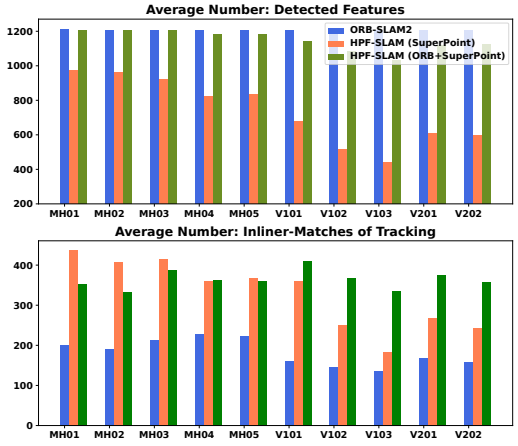


Fig. 4. Comparison on the number of detected features (above) and inliner matches (below) with EuRoc-MAV. Using hybrid point features and the proposed matching scheme, we achieve significantly more correct matches.

results show that HPF-SLAM (ORB+SuperPoint) achieves the best accuracy in 6 out of the 10 sequences, indicating its competitiveness. Compared to HPF-SLAM (SuperPoint) and the baseline ORB-SLAM2, HPF-SLAM (ORB+SuperPoint) brings improvements in 7 sequences, reflecting the complementary advantages. When using HPF-SLAM (SuperPoint), the tracking exhibits improvements in 6 sequences, while it fails in difficult sequence v103, which can be explained by the performance limitation of SuperPoint (insufficient features, as shown in Figure 4).

Results on TUM-RGBD We present the results on the TUM+RGBD dataset in Table II. Compared to baseline (ORB-SLAM2), HPF-SLAM (ORB+SuperPoint) improves 6 out of 7 sequences, consistently demonstrating the complementary advantages. The method also shows competitive results with the best accuracy in 4 sequences. Tracking with pure SuperPoint encountered failures in 2 sequences (fr1_desk and fr_desk2), which involve sudden and rapid motion. Despite this, the complementary advantages are still observable in sequences fr_desk and V103.

C. Extraction and Matching Performance

The count of correct matches established from feature points is critical to system accuracy and robustness. In Figure 4, we compare the average number of detected features and inliner matches with the EuRoc-MAV dataset. In the case of ORB-SLAM2 with 1200 features, the system detects a sufficient number of ORB features, but only a smaller portion can be accurately associated. When using solely SuperPoint, the number of detected features is limited, particularly in challenging sequences (MH04, V103, V202). We explain this pattern by the sensitivity of SuperPoint to motion blur. Interestingly, despite fewer features, we still acquire more matches than the ORB case, likely due to the repetitiveness of SuperPoint. In the case of using 600 ORB + 600 SuperPoint, we establish significantly more feature pairs.

TABLE I

TRACKING ERROR RMS-ATE (M) ON EUROC-MAV DATASET, WITH LOOP-CLOSURE.

Sequences	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202
VINS-Mono [27]	0.120	0.120	0.130	0.180	0.210	0.068	0.084	0.190	0.081	0.160
DF-SLAM [28]	0.037	0.043	0.046	0.063	**0.042**	0.086	0.064	0.065	0.058	0.058
Attention-SLAM [29]	0.045	0.034	0.037	**0.057**	0.047	0.095	0.063	0.102	0.058	**0.057**
LIFT [30]	0.044	0.053	0.049	-	-	0.157	-	-	-	-
OV^2-SLAM [3]	0.040	0.040	0.040	0.060	0.070	0.090	0.070	0.090	0.070	0.060
Baseline (ORB-SLAM2)	0.030	0.038	0.049	0.178	0.094	0.039	0.038	0.064	0.064	0.090
HPF-SLAM (SuperPoint)	0.028	**0.023**	0.039	0.212	0.094	**0.035**	0.032	X	**0.057**	0.099
HPF-SLAM (ORB+SuperPoint)	**0.025**	**0.023**	**0.030**	0.167	0.116	**0.035**	**0.030**	**0.053**	0.060	0.090

TABLE II

TRACKING ERROR RMS-ATE (M) ON TUM-RGBD DATASET, WITH LOOP-CLOSURE.

Sequences	fr1_xyz	fr1_desk	fr1_desk2	fr2_xyz	fr2_desk	fr3_office	fr3_nst
RGBD-SLAM2 [31]	0.014	0.026	0.025	0.026	0.057	-	-
ElasticFusion [32]	0.016	0.020	0.048	0.011	0.071	0.017	0.018
Kintinuous [33]	0.018	0.037	0.071	0.029	0.034	0.030	0.031
SM-SLAM[34]	0.010	-	-	**0.002**	0.015	0.026	0.068
Fu *et al*. [35]	0.011	0.020	**0.009**	0.007	**0.009**	0.018	0.021
Baseline (ORB-SLAM2)	0.010	0.022	0.029	0.005	0.017	0.036	0.035
HPF-SLAM (SuperPoint)	**0.009**	X	X	0.005	0.016	0.023	**0.017**
HPF-SLAM (ORB+SuperPoint)	**0.009**	**0.019**	0.035	0.004	0.011	**0.013**	0.030

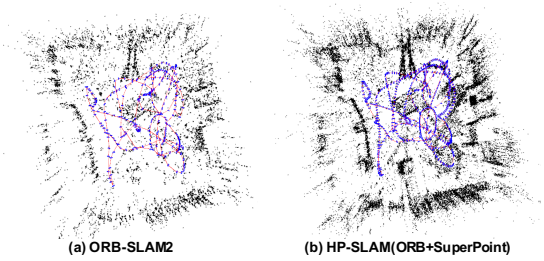


Fig. 5. Visualization of map generated from sequence V102, EuRoc-MAV: ORB-SLAM2 (left), HPF-SLAM (ORB+SuperPoint) in right.

TABLE III

FRAME LEVEL RUN-TIME EVALUATION.

Dataset	EuRoc-MAV		TUM-RGBD	
Camera Type	Stereo Camera		RGBD Camera	
Image Resolution	752×480		640×480	
Frame Rate	20		30	
Run-Time (ms)	mean	std	mean	std
Pre-Processing	27.27	1.39	9.71	1.83
Pose Prediction	1.75	0.71	3.05	0.85
Track LocalMap	3.49	1.46	5.16	0.91
Total Processing Time	34.09	3.29	19.47	2.61

D. Comparison on Maps

In Figure 5 we visualize the maps reconstructed from sequence V102 of the EuRoc-MAV dataset. Compared to ORB-SLAM2 (left), HPF-SLAM (ORB+SuperPoint) reconstructs a much denser map with significantly more landmarks. With more environmental information, the denser map can further benefit the map reuse. Due to more keyframes and map-points, the map size in storage is larger than in ORB-SLAM2. Despite this, it's much smaller than in other learnable systems with deep (float) descriptors [10], as the 2D and 3D features in the map are with binarized descriptors.

E. Run-Time Evaluation

In Table III, we present the run-time evaluation of HPF-SLAM (ORB+SuperPoint). The experimental results show that the tracking operations in HPF-SLAM maintain real-time performance on both datasets. Since the EuRoc-MAV dataset is equipped with a stereo camera, the pre-processing time is longer for additional processes such as stereo-matching. Despite this, the proposed framework still works in real-time at the camera frame level.

V. CONCLUSIONS

In this work, we presented HPF-SLAM, an efficient system leveraging both hand-crafted and learnable features for visual SLAM. We designed a real-time pre-processing module to acquire the hybrid point features. We further introduced an efficient matching scheme to process these hybrid point features and address the cross-class mismatching problem. Through extensive evaluation, we presented the complementary advantages, competitiveness, and efficiency of HPF-SLAM. In future work, we aim to improve the versatility of the system by integrating new features. We propose improving the fusion scheme of hybrid point features. We also plan to explore Semantic SLAM by incorporating respective information of learnable features.

REFERENCES

[1] Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-Scale Direct Monocular SLAM. In Proceedings of the Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014; pp. 834–849.

[2] Engel, J.; Koltun, V.; Cremers, D. Direct Sparse Odometry. IEEE Trans. Pattern Anal. Mach. Intell. 2018, 40, 611–625.

[3] M. Ferrera, A. Eudes, J. Moras, M. Sanfourche and G. Le Besnerais, OV^{2}SLAM: A Fully Online and Versatile Visual SLAM for Real-Time Applications, in IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 1399-1406.

[4] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[5] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, ORB: An efficient alternative to SIFT or SURF, in 2011 International Conference on Computer Vision, Nov. 2011, pp. 2564–2571.

[6] D. DeTone, T. Malisiewicz, and A. Rabinovich, SuperPoint: Self-Supervised Interest Point Detection and Description, in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 337–33712.

[7] Revaud, Jérôme, César Roberto de Souza, M. Humenberger and Philippe Weinzaepfel. R2D2: Reliable and Repeatable Detector and Descriptor. Neural Information Processing Systems (2019).

[8] Q. Zhou, T. Sattler, and L. Leal-Taixe, Patch2Pix: Epipolar-Guided Pixel-Level Correspondences, in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA: IEEE, Jun. 2021, pp. 4667–4676.

[9] 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/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 12 716–12 725.

[10] D. Li et al., DX-SLAM: A Robust and Efficient Visual SLAM System with Deep Features, in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA: IEEE, Oct. 2020, pp. 4958–4965.

[11] J. Tang, L. Ericson, J. Folkesson, and P. Jensfelt, "GCNv2: Efficient correspondence prediction for real-time SLAM," IEEE Robotics and Automation Letters, vol. 4, no. 4, pp. 3505–3512, 2019.

[12] X. Wang, Z. Liu, Y. Hu, W. Xi, W. Yu, and D. Zou, "FeatureBooster: Boosting Feature Descriptors with a Lightweight Neural Network." arXiv, Nov. 28, 2022. Accessed: Jan. 23, 2023. [Online]. Available: http://arxiv.org/abs/2211.15069

[13] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning Feature Matching With Graph Neural Networks," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA: IEEE, Jun. 2020, pp. 4937–4946.

[14] D. Galvez-López and J. D. Tardos, "Bags of Binary Words for Fast Place Recognition in Image Sequences," in IEEE Transactions on Robotics, vol. 28, no. 5, pp. 1188-1197, Oct. 2012.

[15] S. Eger, R. Pries, and E. Steinbach, "Evaluation of Different Task Distributions for Edge Cloud-based Collaborative Visual SLAM," in 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP), Sep. 2020, pp. 1–6.

[16] I. Abaspur Kazerouni, L. Fitzgerald, G. Dooly, and D. Toal, "A survey of state-of-the-art on visual SLAM," Expert Systems with Applications, vol. 205, p. 117734, Nov. 2022.

[17] S. Mokssit, D. B. Licea, B. Guermah, and M. Ghogho, "Deep Learning Techniques for Visual SLAM: A Survey," IEEE Access, vol. 11, pp. 20026–20050, 2023.

[18] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," in IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255-1262, Oct. 2017.

[19] Z. Teed and J. Deng. "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras." Advances in Neural Information Processing Systems 34 (2021): 16558-16569.

[20] Q. Fu et al., "Fast ORB-SLAM without Keypoint Descriptors," IEEE Trans. on Image Process., vol. 31, pp. 1433–1446, 2022.

[21] Y. Qiu, C. Wang, W. Wang, M. Henein, and S. Scherer, "AirDOS: Dynamic SLAM benefits from Articulated Objects," in 2022 International Conference on Robotics and Automation (ICRA), May 2022, pp. 8047–8053.

[22] K. Xu, Y. Hao, S. Yuan, C. Wang, and L. Xie, "AirVO: An Illumination-Robust Point-Line Visual Odometry." arXiv, Aug. 04, 2023.

[23] B. Zi, H. Wang, J. Santos and H. Zheng, "An Enhanced Visual SLAM Supported by the Integration of Plane Features for the Indoor Environment," 2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN), Beijing, China, 2022, pp. 1-8.

[24] Yang H, Yuan J, Gao Y, et al. UPLP-SLAM: Unified point-line-plane feature fusion for RGB-D visual SLAM[J]. Information Fusion, 2023, 96: 51-65.

[25] M. Burri et al.,"The EuRoC micro aerial vehicle datasets," The International Journal of Robotics Research, vol. 35, no. 10, Sep. 2016, pp. 1157-1163.

[26] Sturm J, Engelhard N, Endres F, et al. A benchmark for the evaluation of RGB-D SLAM systems[C]//2012 IEEE/RSJ international conference on intelligent robots and systems. IEEE, 2012: 573-580.

[27] 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, Aug. 2018.

[28] R. Kang, J. Shi, X. Li, Y. Liu, and X. Liu, "DF-SLAM: A Deep-Learning Enhanced Visual SLAM System based on Deep Local Features." arXiv, Jan. 24, 2019. doi: 10.48550/arXiv.1901.07223.

[29] J. Li et al., "Attention-SLAM: A Visual Monocular SLAM Learning From Human Gaze," IEEE Sensors Journal, vol. 21, no. 5, pp. 6408–6420, Mar. 2021, doi: 10.1109/JSEN.2020.3038432.

[30] H. M. S. Bruno and E. L. Colombini, "LIFT-SLAM: a deep-learning feature-based monocular visual SLAM method," Neurocomputing, vol. 455, pp. 97–110, Sep. 2021, doi: 10.1016/j.neucom.2021.05.027.

[31] Endres F, Hess J, Sturm J, et al. 3-D mapping with an RGB-D camera[J]. IEEE transactions on robotics, 2013, 30(1): 177-187.

[32] T. Whelan, RF. Salas-Moreno, B. Glocker, et al., " ElasticFusion: Real-time dense SLAM and light source estimation," Int. J. Robot. Research, vol. 35, no. 4, pp. 1697-1716, 2016.

[33] Whelan, T., Kaess, M., Fallon, M., Johannsson, H., Leonard, J., McDonald, J. (2012). Kintinuous: Spatially extended kinectfusion.

[34] H. Xie et al., Semi-Direct Multimap SLAM System for Real-Time Sparse 3-D Map Reconstruction, IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1–13, 2023.

[35] Q. Fu , H. Yu, L. Lai, et al., "A Robust RGB-D SLAM System With Points and Lines for Low Texture Indoor Environments," IEEE Sens. J., vol. 19, no. 1, pp. 9908-9920, 2019.