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

EdgePoint: Efficient Point Detection and Compact Description via Distillation

Haodi Yao[1], Ning Hao[1], Chen Xie[1], Fenghua He[1,†]

***Abstract*— Efficient interest point detection and description in images play a crucial role in many tasks such as multi-robot SLAM and collaborative localization. To facilitate fast detection and generate compact descriptions on edge devices, we introduce EdgePoint, a lightweight neural network. We design a new detection loss UnfoldSoftmax to improve inference speed. Futhermore, we propose Ortho-Alignment loss combined with LocalPCA compression to learn compact 32-dimensional descriptors. To enable efficient storage or communication, we also quantize the generated descriptors into integral values. We perform EdgePoint on various datasets, and show that it surpasses SuperPoint in performance while utilizing only 1% of the parameters and achieving up to more than 10 times faster inference speed. By applying descriptor quantization, the requirements for storage and communication can be reduced by up to 97% without performance decreasing.**

I. Introduction

The detection and description of interest points in images are fundamental components of many robotic systems. Research areas such as Simultaneous Localization and Mapping (SLAM) [1], Structure-from-Motion (SfM) [2] are of high interest, where they need points that can be detected and re-identified in a wide range of scenarios including illumination changes and viewpoint changes. Traditionally, these tasks rely on handcrafted features such as SIFT [3], BRIEF [4] or ORB [5], which may limit the performance in challenging situations. With deep learning methods having made breakthroughs in computer vision [6], there is a lot of remarkable work in interest point detection and description [7, 8]. These work have significantly improved the performance of robotic systems.

Despite the improvements brought by deep learning, with the growing interest in multi-robot systems like multi-robot SLAM [9, 10] and collaborative localization [11, 12], these networks encounter difficulties when deployed on on-board devices with limited computational resources, making real-time processing challenging. Besides, robots may be equipped with multiple cameras in some scenarios, requiring much more computational resources [13, 14]. Additionally, these descriptors are high-dimensional in floating-point numbers to represent a single point. The high dimensionality leads to heavy communication burden when transferring these descriptors among robots. Some work adopt principal component analysis (PCA) to compress the descriptors [13, 14]. However, this approach does not accelerate the network inference and requires data collected in advance for PCA. Thus, there is an urgent demand for fast point detection and compact description methods to mitigate the communication burden between robots with onboard edge devices.

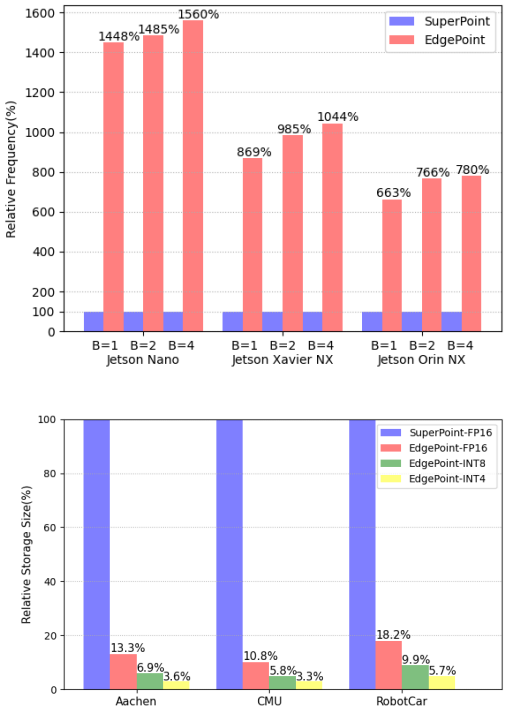

Fig. 1. **Relative Speed and Storage Test Results.** EdgePoint can run faster up to 16× than SuperPoint on edge devices while costing around 4% of the storage size

In this paper, we design an extremely small neural network, EdgePoint, for edge devices to run in real-time even with multiple cameras. In this work, we use SuperPoint for detection and description distillation. We simplify the detection block to avoid high cost computation with our newly designed UnfoldSoftmax loss function. Additionally, in order to acquire compact descriptors with high performance, we propose a LocalPCA method to compress descriptors in single image rather than previous whole dataset compression. Furthermore, the Ortho-Alignment loss is proposed to solve the mismatch caused by compressing individually. We also adopt a descriptor quantization method for the purpose of further reducing the cost of communication and storage. The relative speed and storage comparison are shown in Fig. 1.

[1]All authors are with the School of Astronautics, Harbin Institute of Technology, Harbin, China.
[†]Corresponding Author (Email: hefenghua@hit.edu.cn)

To summarize, the main contributions of this work are as follows:

- We design a tiny model, EdgePoint, for fast interest point detection and compact description which can achieve about 1ms inference latency on edge devices.
- We propose an UnfoldSoftmax detection loss to accelerate the point detection in inference stage.
- We compress descriptors of each image and propose an Ortho-Alignment loss function to achieve distillation.
- We provide a quantization method for compact descriptors to further reduce the requirements of communication and storage.
- Experiments on various datasets demonstrate the high efficiency and performance of EdgePoint.

II. Related Work

Traditional interest point detectors, such as Shi-Tomasi [15] and FAST [16], continue to play an important role in robotic systems when combined with handcrafted descriptors like SIFT [3] and ORB [5]. In recent years, deep learning has emerged, leading to numerous studies on robust point detection and invariant description [17]–[19]. While some researchers have treated point detection as a classification problem [7], others have pursued sub-pixel point detection through regression-based methods [20, 21]. For local descriptors, there have been efforts to generate higher-quality representations using techniques like contrastive learning [22] and self-supervised learning [7]. Additionally, while most of these approaches utilize Structure-from-Motion (SfM) data for training [8], the use of data with homographic enhancement has also been explored to simplify and widen data preparation [7, 20, 21]. Consequently, these advancements have significantly enhanced the performance of robotic applications, particularly in challenging scenarios characterized by large-scale viewpoint changes or illumination variations.

However, implementing neural networks especially for large models on robots with onboard computers presents challenges due to limited resources, making real-time execution challenging. In response, numerous studies have focused on improving efficiency and reducing parameters, leading to the development of lightweight model architectures such as MobileNet [23], EfficientNet [24] and FasterNet [25]. Efforts have been made to distill interest point detection and description networks into lightweight model architectures [26]. Model quantization are also investigated in [27] to accelerate inference in compute limited platforms.

In addition, with the increasing demand for multi-robot or multi-camera applications, the urgency for compact local descriptors has grown. While classic approaches commonly use 256-dimensional descriptors, such as SIFT and SuperPoint [3, 7], compact descriptors have been proposed in [8]. However, even these compact descriptors are still demanding in terms of communication requirements. An alternative approach is to use binary descriptors [4, 27], which can reduce the communication burden, but may suffer from performance limitations compared to floating-point descriptors. Another technique is using PCA to compress descriptors for communication and storage [13, 14]. Despite these advancements, there is still a demand for faster, lighter neural networks that can directly generate more compact descriptors.

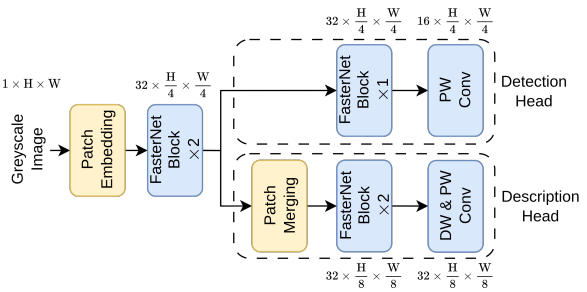

Fig. 2. **EdgePoint Model Architecture**

III. Methodology

In this section, we start by introducing the overall network architecture backbone designed specifically for EdgePoint. Then, we propose an UnfoldSoftmax detection loss that aims to accelerate interest point detection in inference stage. To achieve lower dimensionality, LocalPCA is employed for local descriptors compression. We propose an Ortho-Alignment loss for descriptors distillation. Furthermore, a descriptor quantization method is designed to further reduce the requirements of communication and storage. Finally, we present the training technique along with the implementation details.

A. Model Architecture

As illustrated in Fig. 2, EdgePoint consists of three components: a shared encoder, a point decoder, and a descriptor decoder. We utilize the FasterNet block [25] with ReLU activation as the basic building block for high inference efficiency with higher FLOPs and less parameters. The patch embedding layer is a 4×4 convolution layer with a stride of 4, while the merging layer is a 2×2 convolution layer with a stride of 2. In order to maintain point detection performance, we choose 32 channels for the first layer and adopt a downsampling factor of 4 for the feature map. For efficiency, we directly use the raw output instead of the Softmax result for detection before pixel shuffling. This is mainly because the Softmax operator may significantly increase inference latency, especially on edge devices. As for description, we maintain 32 channels and downsample the feature map to $1/8$ size to achieve faster inference speed. The descriptors are L2 normalized and bilinearly sampled to ensure accurate representations.

B. Detection Distillation

Previous approaches commonly treat interest point detection as a classification problem using Sigmoid or Softmax. Sigmoid enables the detection of each point while this leads to a noticeable decrease of detection performance in our shallow network design. Alternatively, Softmax represents

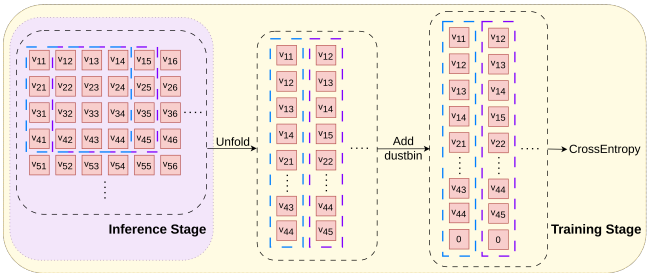

Fig. 3. **Training and Inference with UnfoldSoftmax**

a patch of pixels and requires an extra channel to represent the "dustbin" class. Although it is possible to remove this step during inference by utilizing the raw output, the results may be inconsistent across blocks. This inconsistency arises due to the Softmax being calculated independently in each non-overlapping grid region, and the non-maximum suppression(NMS) being applied across these regions.

To overcome these limitations, we propose a method where the output is directly generated without any activation during inference. In the training stage, the feature map is unfolded into vectors of size $k \times k$ with a stride of 1. We then append 0 to each vector as a dustbin element. Then the cross entropy is calculated. The procedure is demonstrated in Fig. 3. We refer to the loss as UnfoldSoftmax, which can be expressed as

$$L_{\text{detect}} = \sum_{p}^{P} \text{UnfoldSoftmax}\left(X_p; Y_p\right), \tag{1}$$

where P represents all the patches that can be extracted from a given image, X_p and Y_p correspond to the detection output and its corresponding label for the patch, respectively. The label Y_p is a binary vector of size $k \times k$, where each element can take on the values of either 0 or 1. If an interest point does not exist in a particular patch, we append 1 to the unfolded vector to represent the absence of an interest point. In practice, we set $k = 9$ for the detection distillation loss, corresponding to an NMS radius of 4.

C. Descriptor Compression and Distillation

While SuperPoint generates descriptors with 256 dimensions, low-dimensional descriptors are typically sufficient for feature matching in practice. A common approach is compressing the descriptors using PCA before post-processing or communication [13, 14]. However, this involves performing statistical analysis on the entire designated dataset. As the dataset volume increases, the information loss from PCA compression becomes more significant. To ensure minimal compression loss on the interest point descriptors within each image, we propose to perform compression on each image individually. This approach allows us to optimize the compression process as each detected interest point can represent one piece of data. We refer to this approach as LocalPCA.

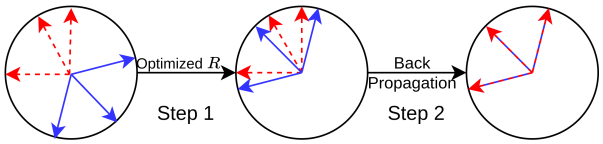

Fig. 4. **Illustration of Ortho-Alignment Loss**. The blue arrow and red arrow represent teacher descriptors and student descriptors respectively.

We define generated N descriptors for an image as $D = [d_1, d_2, ..., d_N]^{\mathrm{T}} \in \mathbb{R}^{N \times C}$, where C is the descriptor dimension. These descriptors are L2-normalized, meaning that each descriptor $d_i \in \mathbb{R}^C$ satisfies $\|d_i\|_2 = 1$.

Denoting D_t and D_s as the outputs of the teacher network and student network, respectively, we can express the previous descriptor distilling loss as

$$L_{\cos} = \|\mathbf{1} - \text{diag}(D_s D_t^{\mathrm{T}})\|_1, \tag{2}$$

where $\mathbf{1}$ represents an all ones vector.

However, due to the separate compression of descriptors for each image, it is not possible to preserve distance measure cross images. Thus we propose a novel Ortho-Alignment loss for descriptor distillation task by seperating the original matching loss into the absolute loss and the relative loss components. Assuming that the data quantity satisfies $N > C$, by introducing an additional matrix R into Eq. (2), the Ortho-Alignment loss function can be represented as

$$L_{\text{align}} = \|\mathbf{1} - \text{diag}(D_s R D_t^{\mathrm{T}})\|_1, \tag{3}$$

where $R \in \mathbb{O}(C, \mathbb{R}) = \{X \in \text{GL}(C, \mathbb{R}) | X^{\mathrm{T}} X = X X^{\mathrm{T}} = I\}$ represents an orthogonal matrix, the $\mathbb{O}$ is the orthogonal group and the GL refers to the general linear group.

The Ortho-Alignment loss in Eq. (3) uses matrix R as the optimal orthogonal mapping for coarse alignment of the descriptors between the student network and the teacher network. Then the traditional neural network optimizer is used for fine optimization. The whole procedure is shown in Fig. 4. To minimize L_{align} by optimizing the orthogonal matrix R, we have

$$R = \underset{X \in \mathbb{O}(C, \mathbb{R})}{\arg\min} \ \|\mathbf{1} - \text{diag}(D_s X D_t^{\mathrm{T}})\|_1. \tag{4}$$

Using manifold optimization to solve the optimal solution of Eq. (4) is computationally intensive and time-consuming, especially since it must be executed for each image in every epoch. Considering any $d_i, d_j \in \mathbb{R}^C$, $\|d_i\|_2 = \|d_j\|_2 = 1$, we have

$$1 - d_i^{\mathrm{T}} X d_j \in [0, 2], \forall X \in \mathbb{O}(C, \mathbb{R}). \tag{5}$$

Thus we can turn Eq. (4) into the following form:

$$\begin{aligned} R &= \underset{X\in\mathbb{O}(C,\mathbb{R})}{\arg\min} \ \|\mathbf{1} - \operatorname{diag}(D_s X D_t^{\mathrm{T}})\|_1 \\ &= \underset{X\in\mathbb{O}(C,\mathbb{R})}{\arg\min} \ (N - \operatorname{tr}(D_s X D_t^{\mathrm{T}})) \\ &= \underset{X\in\mathbb{O}(C,\mathbb{R})}{\arg\max} \ \operatorname{tr}(D_s X D_t^{\mathrm{T}}) \\ &= \underset{X\in\mathbb{O}(C,\mathbb{R})}{\arg\max} \ \operatorname{tr}(D_t^{\mathrm{T}} D_s X). \end{aligned} \tag{6}$$

Then the optimal solution for R can be obtained through the Singular Value Decomposition (SVD) as

$$\begin{aligned} U, \Sigma, V^{\mathrm{T}} &= \operatorname{SVD}(D_t^{\mathrm{T}} D_s) \\ R &= V U^{\mathrm{T}}, \end{aligned} \tag{7}$$

where $U, V \in \mathbb{O}(C, \mathbb{R})$, Σ is a $C \times C$ diagonal matrix with non-negative numbers on the diagonal.

By solving Eq. (6), we obtain the optimal orthogonal matrix R for each single image. Then we can perform back propagation using the network optimizer.

Furthermore, for any $d_i, d_j \in \mathbb{R}^C$, $\|d_i\|_2 = \|d_j\|_2 = 1$, we have

$$1 - d_i^{\mathrm{T}} d_j = \frac{1}{2}\|d_i - d_j\|_2^2. \tag{8}$$

Thus, ignoring the coefficient, the Eq. (3) in batch form can be written as

$$L_{\text{desc}} = \frac{1}{M} \sum_{b=1}^{B} \|D_{b,s} - D_{b,t} R_b^{\mathrm{T}}\|_F^2, \tag{9}$$

where B is the batch size and $D_{b,s}$, $D_{b,t}$ represent the student descriptors and teacher descriptors of b-th image in batch respectively, M is the total number of the descriptors in a batch of images, $\|\cdot\|_F$ is the Frobenius norm. In training stage, these descriptors are sampled by the detection result of SuperPoint, and we choose $C = 32$ to balance performance and descriptor size.

D. Descriptor Quantization

With the compact descriptors obtained from distillation, it would be beneficial to alleviate the burden of multi-robot communication or storage. Therefore, we quantify the descriptors into INT8 or even INT4 representations through method expressed as follows:

$$d_q = \lfloor q_{\max} \frac{d}{\|d\|_\infty} \rceil, \tag{10}$$

where d and d_q represents the descriptor and its corresponding quantized descriptor respectively, $q_{\max}$ denotes the maximum integer value for the designated quantization, the symbol $\lfloor\cdot\rceil$ represents the round function and $\|\cdot\|_\infty$ represents the infinity norm of a vector.

Using the given Eq. (10), the descriptors can be quantified for the purposes of storage or communication. During descriptor matching, they can be dequantized simply by utilizing L2 normalization.

E. Implementation Details

We utilize the Google Landmark Dataset v2 (GLDv2) [28] for the distillation process. The images are resized to dimension of 240×320, and the official pretrained SuperPoint is employed to generate points and descriptors. These descriptors are compressed into 32 dimensions using LocalPCA. To ensure stability, only images with a minimum of 128 points are included for training, resulting in a final dataset comprising 3.5 million images. To accelerate network training, the compressed descriptors are cached along with the corresponding detections.

The overall distillation loss is calculated as follow:

$$L = L_{\text{detect}} + L_{\text{desc}}. \tag{11}$$

The training is conducted under a batch size of 512 in half precision on a single GPU with 48GB of memory. We employ the AdamW optimizer with an initial learning rate of 0.002. The learning rate is reduced by half after each epoch. The training process is stopped after 5 epochs, which takes approximately 10 hours to complete.

IV. Experiments

In this section, we will first showcase the computational resources utilized by EdgePoint and highlight its impressive inference speed on edge devices. Then, we will proceed experiments on HPatches to evaluate the performance of the EdgePoint for interest point detection and description. Following that, we will evaluate the performance of EdgePoint in visual localization and visual odometry tasks using multiple datasets. Additionally, we will assess the computational resources required by EdgePoint on various edge devices. To compare the influence of the descriptor quantization, we select INT8 and INT4 quantization for standards.

For the detection process, we set a threshold of -2.5 for the raw output and a radius of 4 for NMS. SuperPoint serves as the baseline since EdgePoint is distilled from it. All experiments utilize the default settings provided by SuperPoint. These settings are consistent across all tests.

A. System Runtime

We compare the parameters amount, the computational resources required and the inference speed on edge devices.

Setup

To assess the computational resources, we measure the FLOPs and memory usage of EdgePoint under single precision. The inference speed test is conducted on Jetson Nano, Xavier NX and Orin NX, which are widely used edge devices commonly employed as onboard computers for robots or UAVs. When conducting tests on these edge devices, we utilize TensorRT to perform inference frequency tests using half precision. This approach effectively doubles the inference speed while maintaining the desired precision level, making it the preferred choice for deploying neural networks on Jetson devices. We incorporate a warm-up period of 10 seconds and run 2,000 iterations to gather accurate statistics. We set a batch size of 1, 2, 4 to imitate the performance of

TABLE I

COMPUTATION RESOURCES COMPARISON

			SuperPoint	EdgePoint	Ratio
Params			1,300k	30k	2.32%
FLOPs(G)			26.11	0.36	1.36%
Memory(MB)			474.38	108.13	22.79%
Desc. Dimension			256	32	12.5%
GPU Infer. Freq. (FPS)	Nano	Batch=1	8.3	120.2	1448%
		Batch=2	4.1	60.9	1485%
		Batch=4	2.0	31.2	1560%
	Xavier NX	Batch=1	65.4	569.1	870%
		Batch=2	33.2	327.1	985%
		Batch=4	16.7	174.4	1044%
	Orin NX	Batch=1	124.2	823.3	663%
		Batch=2	61.6	472.1	766%
		Batch=4	30.9	240.9	780%

TABLE II

RESULTS OF HPATCHES HOMOGRAPHY ESTIMATION

Method	Rep.	Loc.	Cor-1	Cor-3	Cor-5	M.Score
SuperPoint	**0.603**	**1.089**	0.455	0.774	0.855	**0.422**
EdgePoint	0.571	1.105	**0.490**	**0.781**	**0.883**	0.389
INT8 Desc.	0.571	1.105	0.488 (-0.002)	0.776 (-0.005)	0.885 (+0.002)	0.389
INT4 Desc.	0.571	1.105	0.483 (-0.007)	0.791 (+0.010)	0.879 (-0.004)	0.389

our model for different number of cameras. This approach ensures that the measurements captured are consistent and representative of the overall performance. All the evaluations are performed using images resized to 480×640 size.

Results The evaluation results are presented in Table. I. It is evident from the results that EdgePoint utilizes approximately 2% of the parameter amount and 1% of the FLOPS compared to SuperPoint. Moreover, EdgePoint exhibits a significant speed improvement, up to 16 times faster on edge devices. Additionally, EdgePoint proves to be memory-efficient by reducing memory usage by about 80%.

B. Homography Estimation

Next, we use the HPatches [29] dataset to demonstrate the effectiveness of EdgePoint. We will also assess the effects of our quantization method on the descriptors.

Setup The HPatches consists of 116 scenes with each containing 6 images, categorized based on illumination and viewpoint changes. The test is conducted using an image size of 480×640 and a maximum of 1000 points. The benchmark code is based on [20]. For nearest neighbor matching, a distance threshold of 0.7 is set for positive matches. We report repeatability (Rep.), localization error (Loc.) as detector metrics with a correctness distance threshold of 3. Additionally, we present homography accuracy with thresholds of 1, 3, and 5 pixels (Cor-1, Cor-3, Cor-5). The matching score (M.Score) is also provided under a threshold of 3.

Results We report the results in Table. II. Our EdgePoint drops a little bit in detection, but outperforms SuperPoint of Cor-1, Cor-3 and Cor-5 in homography estimation. Furthermore, the quantified descriptors, regardless of being INT8 or INT4, exhibit nearly identical performance to the FP32 descriptors, indicating the success of our quantization strategy in compressing the descriptors without sacrificing performance.

C. Visual Localization

Setup We utilize the Hierarchical Localization framework proposed in [30]. The HLoc framework is designed to support testing on multiple datasets. For our experiments, we select three datasets: the Aachen Day-Night v1.1, the Extended CMU-Seasons and the RobotCar-Seasons v2 [31]. To assess the impact of daytime and nighttime illumination changes, we employ the Aachen Day-Night v1.1 dataset. The Extended CMU-Seasons and the RobotCar-Seasons v2 are used to evaluate localization performance under varying seasonal conditions.

In terms of the hyperparameter settings, we set the distance threshold for nearest neighbor matching to 0.7 for both EdgePoint and SuperPoint. Additionally, during the test, we enable the covisibility clustering. The image size, maximum keypoint number and the other settings are all kept the same as the default settings in the HLoc framework.

As for the evaluation metrics, we set three groups of thresholds containing distance and orientation errors: $(0.25m, 2°)$, $(0.5m, 5°)$, and $(5m, 10°)$. By using these thresholds, we are able to assess the performance of the localization approach at different levels of accuracy. To further demonstrate the effectiveness of our quantization method, we have calculated the size of points and descriptors in terms of gigabytes. The default setting in HLoc saves SuperPoint and EdgePoint features in half precision, while the quantified descriptors are stored as integers.

Results Table. III presents the recall at different position and orientation thresholds. In the case of Aachen v1.1, EdgePoint achieves comparable performance as SuperPoint in Day scenarios while slightly outperforming it in Night scenarios. In Extended CMU Seasons, EdgePoint achieves similar performance to SuperPoint in Urban and Suburban scenarios, with a maximum drop of 2.6% in Park scenarios. In RobotCar Season v2 dataset, EdgePoint performs equally well as SuperPoint in Day scenarios and show improvement of up to 10% in Night scenarios. Furthermore, when comparing the FP16 descriptors with the INT8 and INT4 quantized descriptors, the results show almost no performance loss in every dataset for our EdgePoint. In some metrics, the quantized descriptors even outperform SuperPoint.

In terms of feature sizes, EdgePoint already provides a memory saving of around **85**% comparing to SuperPoint. By utilizing descriptor quantization, the storage usage can be further reduced to **3**% without any noticeable loss in performance. This improvement not only enables more efficient storage of features but also reduces communication requirements for multi-robot applications. Overall, for the visual localization task, EdgePoint demonstrates slightly superior performance compared to SuperPoint under a much smaller descriptor dimension and much less storage consumption.

TABLE III

RESULTS OF HLOC VISUAL LOCALIZATION

Method	Aachen Day-Night v1.1		Feats.	Extended CMU-Seasons			Feats.	RobotCar Seasons v2		Feats.
	Day	Night	(GB)	Urban	Suburban	Park	(GB)	Day all	Night all	(GB)
	(0.25m, 2°) / (0.5m, 5°) / (5m, 10°)			(0.25m, 2°) / (0.5m, 5°) / (5m, 10°)				(0.25m, 2°) / (0.5m, 5°) / (5m, 10°)		
SuperPoint	87.5 / **94.2** / 98.2	64.4 / 83.2 / 95.3	10.5	**92.9** / **96.1** / **98.1**	**85.4** / **89.5** / **93.8**	**73.7** / **78.5** / **83.2**	115	**64.7** / 94.2 / **99.3**	11.4 / 21.0 / 31.7	19.2
EdgePoint	**88.3** / **94.2** / **98.4**	**66.5** / **84.3** / **96.9**	**1.4**	92.3 / 95.4 / 97.6	84.4 / 88.7 / **93.8**	71.1 / 76.3 / 82.2	**12.4**	64.5 / **94.3** / 99.2	**14.7** / **29.6** / **41.5**	**3.5**
INT8 Desc.	88.5 / 94.3 / 97.8 (+0.2/+0.1/-0.6)	66.5 / 86.4 / 96.9 (+0.0/+2.1/+0.0)	0.72	92.3 / 95.4 / 97.6 (+0.0/+0.0/+0.0)	84.6 / 88.8 / 93.9 (+0.2/+0.1/+0.1)	71.2 / 76.5 / 82.2 (+0.1/+0.2/+0.0)	6.64	64.5 / 94.3 / 99.3 (+0.0/+0.0/+0.1)	12.8 / 26.3 / 38.9 (-1.9/-3.3/-2.6)	1.9
INT4 Desc.	88.5 / 94.1 / 98.4 (+0.2/-0.1/+0.0)	66.5 / 84.8 / 96.3 (+0.0/+0.5/-0.6)	∼0.38	92.1 / 95.1 / 97.5 (-0.2/-0.3/-0.1)	84.1 / 88.5 / 93.7 (-0.3/-0.2/-0.1)	70.6 / 76.1 / 82.0 (-0.5/-0.2/-0.2)	∼3.76	64.5 / 94.3 / 99.3 (+0.0/+0.0/+0.1)	12.6 / 26.3 / 38.9 (-2.1/-3.3/-2.6)	∼1.1

* ∼ indicates an estimated size as INT4 is not a standardized data type.

TABLE IV

RMSE RESULTS FOR VINS FUSION ON EUROC

	Method	MH-01	MH-02	MH-03	MH-04	MH-05	V1-01	V1-02	V1-03	V2-01	V2-02	V2-03
Mono-IMU	Original	0.186	0.088	**0.135**	**0.193**	0.354	0.063	0.086	0.160	0.059	-	-
	SuperPoint	0.186	0.101	0.213	0.215	**0.232**	**0.053**	0.071	**0.085**	0.085	-	**0.182**
	EdgePoint	**0.109**	**0.078**	0.200	0.502	0.256	**0.053**	**0.060**	0.103	**0.058**	**0.321**	0.554
Stereo	Original	0.571	0.510	**0.490**	0.843	0.618	0.523	0.232	2.724	0.323	0.219	-
	SuperPoint	0.483	0.434	0.497	0.634	**0.474**	0.143	0.114	**0.168**	**0.236**	0.201	**1.660**
	EdgePoint	**0.450**	**0.392**	0.545	**0.518**	0.495	**0.136**	**0.104**	1.697	0.363	**0.174**	2.015
Stereo-IMU	Original	0.253	0.217	0.326	0.434	**0.312**	0.113	0.108	0.102	0.131	0.124	0.272
	SuperPoint	0.207	0.207	**0.251**	**0.394**	0.351	0.108	**0.089**	**0.073**	0.096	0.098	**0.122**
	EdgePoint	**0.193**	**0.166**	0.279	0.439	0.338	**0.103**	**0.089**	0.094	**0.089**	**0.096**	0.294

* - indicates divergence of the odometry.

D. Visual Odometry

To demonstrate the effectiveness of EdgePoint, we employ VINS-Fusion [32]–[34], a widely acknowledged onboard visual odometry method.

Setup VINS-Fusion utilizes the Shi-Tomasi method for point detection and the Lucas-Kanade(LK) method [35] for point tracking. Due to the relatively small variation in viewpoint change in visual odometry, a threshold of 0.5 is set for the matching of EdgePoint or SuperPoint descriptors. Additionally, to mimic the local matching policy of the LK method, a distance threshold of 100 pixels is enforced to ensure robust matching. The maximum number of tracking points and other hyperparameters remain the same as the default configuration of VINS-Fusion. To give an impartial results, the loop fusion is not enabled during the test.

We use EuRoC [36] dataset for test as it is specifically designed for micro aerial vehicles, allowing for the evaluation of EdgePoint's performance on robots. It includes stereo images, synchronized IMU measurements and accurate ground trajectories. To comprehensively assess the algorithm's performance in various scenarios, three testing modes are employed: monocular camera with IMU, stereo cameras, and stereo cameras with IMU.

Since visual odometry typically does not require multi-robot communications, quantified descriptors are not tested. The evaluation of the visual odometry results for both datasets is carried out using EVO from [37], with the performance reported in terms of RMSE (Root Mean Squared Error) metrics.

Results The results obtained on the EuRoC dataset are presented in Table. IV. In the case of monocular camera with IMU, the EdgePoint-based odometry outperforms both SuperPoint-based odometry and the original VINS-Fusion by reaching the best metrics in 6 subsets. Notably, the SuperPoint-based odometry diverges in subsets V2-02 and the original VINS-Fusion algorithm also fails in subsets V2-02 and V2-03. In contrast, the EdgePoint-based odometry behaves reliable and has better performance. For the stereo settings, our EdgePoint-based odometry continues to outperform the SuperPoint-based odometry and the original VINS-Fusion in most subsets. Although the original VINS-Fusion diverges in subset V2-03, the EdgePoint-based odometry remains stable. In the case of stereo cameras with IMU, the EdgePoint-based odometry still achieves superior results in the majority of the subsets.

It is worth mentioning that the data used for the experiments is cached. However, the SuperPoint-based odometry may face challenges in terms of frequency and latency when deployed on edge devices. In contrast, our EdgePoint approach will be more competitive in real-time applications.

V. CONCLUSIONS

In this paper, we present EdgePoint, a light-weight interest point detection and description network that is distilled from SuperPoint. We propose UnfoldSoftmax loss specifically designed for our network, eliminating burden during the inference stage. Additionally, we propose a new approach called LocalPCA to compress the descriptions into low dimension and an Ortho-Alignment loss to learn compact descriptions. Our descriptor quantization method further reduces storage or communication usage.

Experimental results show that EdgePoint outperforms SuperPoint across multiple datasets while utilizing only 2% of the parameters, 12.5% of the descriptor dimensions, and 3% of the storage. Moreover, EdgePoint is up to 16 times faster than SuperPoint, making it an efficient and effective solution for multi-camera or multi-robot tasks.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski, "Bundle adjustment in the large," in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II 11*. Springer, 2010, pp. 29–42.

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[4] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*. Springer, 2010, pp. 778–792.

[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*. Ieee, 2011, pp. 2564–2571.

[6] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[7] 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.

[8] M. Tyszkiewicz, P. Fua, and E. Trulls, "Disk: Learning local features with policy gradient," *Advances in Neural Information Processing Systems*, vol. 33, pp. 14 254–14 265, 2020.

[9] Y. Chang, Y. Tian, J. P. How, and L. Carlone, "Kimera-multi: a system for distributed multi-robot metric-semantic simultaneous localization and mapping," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 210–11 218.

[10] Y. Tian, Y. Chang, F. H. Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems," *IEEE Transactions on Robotics*, vol. 38, no. 4, 2022.

[11] T.-K. Chang, K. Chen, and A. Mehta, "Resilient and consistent multirobot cooperative localization with covariance intersection," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 197–208, 2021.

[12] J. Zhu and S. S. Kia, "Cooperative localization under limited connectivity," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1523–1530, 2019.

[13] H. Xu, Y. Zhang, B. Zhou, L. Wang, X. Yao, G. Meng, and S. Shen, "Omni-swarm: A decentralized omnidirectional visual–inertial–uwb state estimation system for aerial swarms," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3374–3394, 2022.

[14] H. Xu, P. Liu, X. Chen, and S. Shen, "D^2 SLAM: Decentralized and distributed collaborative visual-inertial slam system for aerial swarm," *arXiv preprint arXiv:2211.01538*, 2022.

[15] J. Shi *et al.*, "Good features to track," in *1994 Proceedings of IEEE conference on computer vision and pattern recognition*. IEEE, 1994, pp. 593–600.

[16] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*. Springer, 2006, pp. 430–443.

[17] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable cnn for joint detection and description of local features," *arXiv preprint arXiv:1905.03561*, 2019.

[18] J. Revaud, P. Weinzaepfel, C. De Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger, "R2d2: repeatable and reliable detector and descriptor," *arXiv preprint arXiv:1906.06195*, 2019.

[19] Y. Tian, B. Fan, and F. Wu, "L2-net: Deep learning of discriminative patch descriptor in euclidean space," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 661–669.

[20] J. Tang, H. Kim, V. Guizilini, S. Pillai, and R. Ambrus, "Neural outlier rejection for self-supervised keypoint learning," *arXiv preprint arXiv:1912.10615*, 2019.

[21] P. H. Christiansen, M. F. Kragh, Y. Brodskiy, and H. Karstoft, "Unsuperpoint: End-to-end unsupervised interest point detector and descriptor," *arXiv preprint arXiv:1907.04011*, 2019.

[22] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," *Advances in neural information processing systems*, vol. 30, 2017.

[23] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[24] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *International conference on machine learning*. PMLR, 2021, pp. 10 096–10 106.

[25] J. Chen, S.-h. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S.-H. G. Chan, "Run, don't walk: Chasing higher flops for faster neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 021–12 031.

[26] 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.

[27] M. Kanakis, S. Maurer, M. Spallanzani, A. Chhatkuli, and L. Van Gool, "Zippypoint: Fast interest point detection, description, and matching through mixed precision discretization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6113–6122.

[28] T. Weyand, A. Araujo, B. Cao, and J. Sim, "Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2575–2584.

[29] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5173–5182.

[30] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *CVPR*, 2019.

[31] 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.

[32] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," 2019.

[33] T. Qin and S. Shen, "Online temporal calibration for monocular visual-inertial systems," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3662–3669.

[34] 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.

[35] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI'81: 7th international joint conference on Artificial intelligence*, vol. 2, 1981, pp. 674–679.

[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] M. Grupp, "evo: Python package for the evaluation of odometry and slam." https://github.com/MichaelGrupp/evo, 2017.