mathematics

MDPI

Article

OFPoint: Real-Time Keypoint Detection for Optical Flow Tracking in Visual Odometry

Yifei Wang, Libo Sun * and Wenhu Qin

School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China; 220213594@seu.edu.cn (Y.W.); qinwenhu@seu.edu.cn (W.Q.)
* Correspondence: sunlibo@seu.edu.cn; Tel.: +86-25-83795620

Abstract: Visual odometry (VO), including keypoint detection, correspondence establishment, and pose estimation, is a crucial technique for determining motion in machine vision, with significant applications in augmented reality (AR), autonomous driving, and visual simultaneous localization and mapping (SLAM). For feature-based VO, the repeatability of keypoints affects the pose estimation. The convolutional neural network (CNN)-based detectors extract high-level features from images, thereby exhibiting robustness to viewpoint and illumination changes. Compared with descriptor matching, optical flow tracking exhibits better real-time performance. However, mainstream CNN-based detectors rely on the "joint detection and descriptor" framework to realize matching, making them incompatible with optical flow tracking. To obtain keypoints suitable for optical flow tracking, we propose a self-supervised detector based on transfer learning named OFPoint, which jointly calculates pixel-level positions and confidences. We use the descriptor-based detector simple learned keypoints (SiLK) as the pre-trained model and fine-tune it to avoid training from scratch. To achieve multi-scale feature fusion in detection, we integrate the multi-scale attention mechanism. Furthermore, we introduce the maximum discriminative probability loss term, ensuring the grayscale consistency and local stability of keypoints. OFPoint achieves a balance between accuracy and real-time performance when establishing correspondences on HPatches. Additionally, we demonstrate its effectiveness in VO and its potential for graphics applications such as AR.

Keywords: feature extraction; deep learning for machine vision; visual odometry; image registration; visual slam

MSC: 68T07

check for updates

Academic Editor: Junlin Hu

Received: 16 February 2025
Revised: 20 March 2025
Accepted: 25 March 2025
Published: 26 March 2025

Citation: Wang, Y.; Sun, L.; Qin, W. OFPoint: Real-Time Keypoint Detection for Optical Flow Tracking in Visual Odometry. *Mathematics* **2025**, *13*, 1087. https://doi.org/10.3390/math13071087

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Visual odometry (VO) [1] is a key technique for determining motion in 3D reconstruction [2], autonomous driving, and augmented reality (AR) [3]. It serves as the foundation for visual simultaneous localization and mapping (SLAM), enabling pose estimation [4] through keypoint detection and correspondence establishment, where correspondence establishment includes both descriptor matching and optical flow tracking. The VO relies on traditional detectors such as scale-invariant feature transform (SIFT) [5], speeded-up robust features (SURF) [6], and Oriented FAST and rotated BRIEF (ORB) [7] to establish correspondences using high-dimensional descriptors [8], and estimate pose by minimizing reprojection errors [9]. Although descriptor matching establishes robust multi-view geometric constraints between keypoints, the extraction and matching of descriptors affect the real-time performance.

In contrast, while optical flow tracking [10] also requires keypoint detection, it does not rely on descriptors but establishes correspondences based on grayscale patches, exhibiting better real-time performance. In particular, real-time systems like the visual-inertial system (VINS)-Mono [11] and semi-direct visual odometry (SVO) [12] achieve keypoint tracking across consecutive frames using a grayscale patch-based optical flow. If optical flow tracking that balances accuracy and real-time performance is achieved, it can replace descriptor matching for establishing correspondences in VO.

Considering optical flow tracking relies on locations and grayscale patch information of keypoints [13], detecting "optical flow tracking-friendly" keypoints from images is a crucial step for successful tracking in VO. Such keypoints should exhibit repeatability under varying illuminations and viewpoints. Compared to handmade keypoints, convolutional neural network (CNN)-based [14] keypoints demonstrate robustness to illumination variation and can represent higher-level image features. However, mainstream CNN-based detectors [15] establish keypoint correspondences through descriptors, such as SuperPoint (SP) [16] and simple learned keypoints (SiLK) [17], following the "joint detection and description" framework [15] to generate keypoints and descriptors simultaneously.

Due to different methods of establishing correspondences, the keypoints suitable for descriptor matching and those suitable for optical flow tracking are distinct: the former emphasizes keypoint robustness and uniqueness, while the latter prioritizes grayscale consistency and local stability [13]. Consequently, keypoints detected by the "joint detection and description" framework using optical flow tracking may affect the establishment of keypoint correspondences in VO.

The main contributions of this paper, aimed at facilitating optical flow tracking in VO through a keypoint detector named OFPoint, are as follows:

1. To avoid relearning high-dimensional point features for optical flow tracking, we adopt transfer learning [18] by fine-tuning SiLK as the pre-trained network.
2. To ensure scale invariance in keypoint detection, we introduce the multi-scale attention mechanism capturing salient point features at different scales.
3. To promote grayscale consistency and local stability, we propose a loss term by maximizing the distinctiveness between keypoint distances and image grayscale.
4. To validate the effectiveness of the keypoint detector, the VO based on the proposed feature and optical flow tracking is implemented on the KITTI Vision Benchmark Suite (KITTI) [19], and the implementation process of VO is shown in Figure 1.

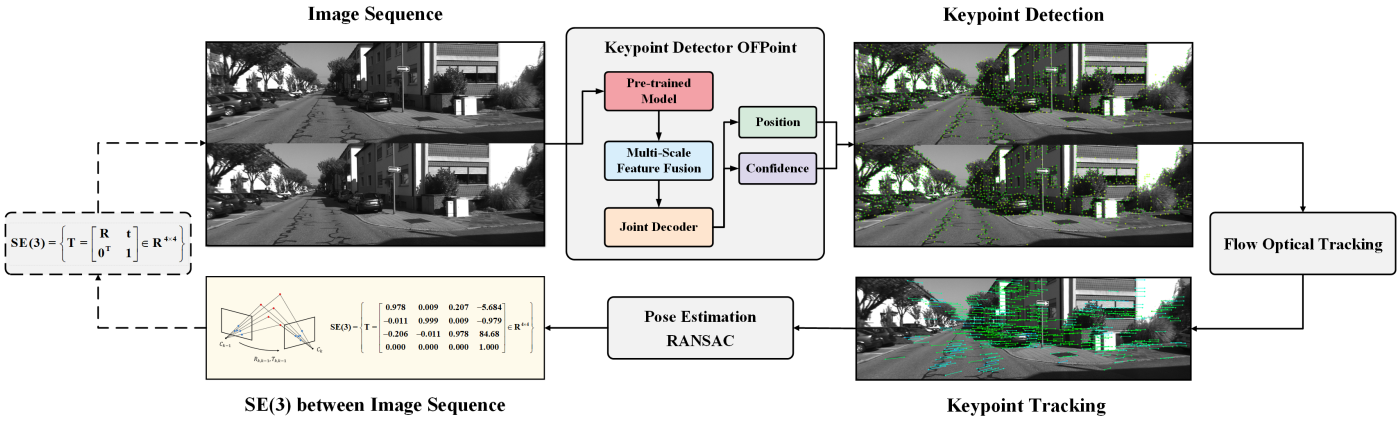


Figure 1. VO consists of three components: keypoint feature detection, correspondence establishment, and pose estimation.

The structure of this paper is as follows: Section 2 provides a detailed introduction to the relevant work, Section 3 presents the specific implementations of the proposed OFPoint architecture, Section 4 covers the training process and the associated loss terms, Section 5 presents the experimental results and analysis, where the performance of OFPoint and its integration with VO are evaluated, and Section 6 offers a summary.

2. Related Work

2.1. Keypoint Detection

VO widely employs traditional keypoint detectors, which are designed to identify keypoints and descriptors in geometric computer vision. Keypoint detectors such as Harris [20], SIFT, ORB, and SURF, are grounded in geometric concepts like corners and gradients. Traditional keypoint detectors follow the "detection-before-description" framework [21], and keypoint matching is achieved by assessing the similarity between high-dimensional descriptors.

Recently, methods utilizing CNN to extract distinctive keypoints and descriptors have been proposed. Learned invariant feature transform (LIFT) (2016) [22] introduced an end-to-end joint formulation consisting of keypoint detection, orientation estimation, and feature description, but its module-wise optimization fails to account for information sharing. To capture information sharing between modules, the "joint detection and description" framework has been proposed, where keypoint detection and descriptor extraction are simultaneously learned and trained within a unified model.

Departing from LIFT, learning local features from images (LF)-Net (2018) [23] leveraged the spatial transformer network for orientation estimation, enabling unsupervised learning of local descriptors. Another example is SuperPoint (2018) [16], which designed a self-supervised framework for simultaneous keypoint detection and descriptor generation. Building upon MagicPoint, which was trained on a synthetic shape dataset with annotated corners, SuperPoint employed multiple homographies on images to detect keypoints under varying viewpoints and scales. Although SuperPoint's self-supervised learning reduces the demand for annotated data, its training process remains complex, requiring the generation of synthetic datasets and multi-stage training.

Unlike SuperPoint, joint description and detection (D2)Net (2019) [24] proposed a single CNN that utilized pixel-level correspondences generated from MegaDepth to train the model, achieving strong coupling between descriptor extraction and feature detection. Building upon D2Net, repeatable and reliable detector and descriptor (R2d2) (2019) [25] introduced a joint measure of reliability and repeatability to guide the optimization. DIScrete Keypoints (DISK) (2020) [26] proposed a learning-based framework for local feature extraction, leveraging policy gradients to enable end-to-end optimization. Rotation-robust descriptors (RoRD) (2021) [27] learned stable descriptors through orthographic projection, addressing extreme viewpoint changes. Accurate and lightweight keypoint detection and descriptor extraction (ALIKE) (2022) [28] introduced a differentiable keypoint detection, which can backpropagate gradients to produce sub-pixel level keypoints and optimize locations via reprojection loss.

FeatureBooster (2023) [29] took raw descriptors and geometric properties of keypoints as input, and employed a self-boosting stage with multi-layer perceptrons and a cross-boosting stage based on Transformers to enhance descriptors. SiLK (2023) [17] proposed a differentiable, lightweight detection-description framework that learned to extract robust keypoints using a self-supervised approach. Despite its simplicity, SiLK achieved state-of-the-art (SOTA) performance on HPatches.

2.2. Optical Flow Tracking

A common problem in computer vision is optical flow tracking, defined as the apparent motion of the pixels between two image frames [30]. Horn and Schunck (HS) [31] leveraged a variational technique to estimate optical flow, formulated as an energy minimization problem. In contrast to HS, Lucas–Kanade (LK) [10] introduced a local constraint and estimated the sparse flow field. To deal with the issue of insufficient pixel displacement, pyramidal Lucas–Kanade (PLK) [32] was proposed to implement a coarse-to-fine strategy. To improve the robustness of illumination, some high-order constancy constraints [33], such as Hessian and Laplacian, were introduced.

Although optical flow tracking offers real-time performance, further optimization is required when deploying it on resource-constrained devices. Improved total variation L1 (TV-L1) [34] optical flow tracking focused on the mathematical formulation of the algorithm and explored a novel optimized implementation of the algorithm from both software and hardware perspectives.

The end-to-end network FlowNet [35] first adopted CNNs to learn the optical flow field from the synthetic annotated Flying Chairs dataset. Following FlowNet, the spatial pyramid network (SPyNet) [36] was formulated based on a coarse-to-fine strategy within variational methods. A convolutional network was trained at each level of the pyramid to compute optical flow. Inspired by iterative updates, recurrent all-pairs field transforms (RAFT) [37] introduced a novel network architecture to improve the efficiency of optical flow tracking. The complexity of end-to-end methods inevitably limits real-time computation. However, by integrating traditional optical flow tracking with other tasks, such as object detection [38], these limitations can be partially alleviated, while maintaining a balance between accuracy and real-time performance within the CNN framework.

Optical flow tracking using OFPoint is neither a traditional optical flow approach nor a learning-based method. Instead, it integrates learning-based optical flow point detection with conventional tracking methods, achieving a balance between computational efficiency and performance. Therefore, we refer to it as hybrid optical flow tracking.

3. Methods

Inspired by the "joint detection and description" framework, we propose a keypoint detector to replace the detector in VO for optical flow tracking, called OFPoint. OFPoint consists of a backbone encoder, multi-scale feature fusion module, and joint decoder. The structure of OFPoint is illustrated in Figure 2. OFPoint generates dense feature maps through its backbone encoder for subsequent processing. The multi-scale feature fusion module and joint decoder produce aligned outputs in a single forward propagation. Since OFPoint's outputs resemble that of traditional detectors, it can replace traditional detection in VO, providing suitable keypoints for optical flow tracking.

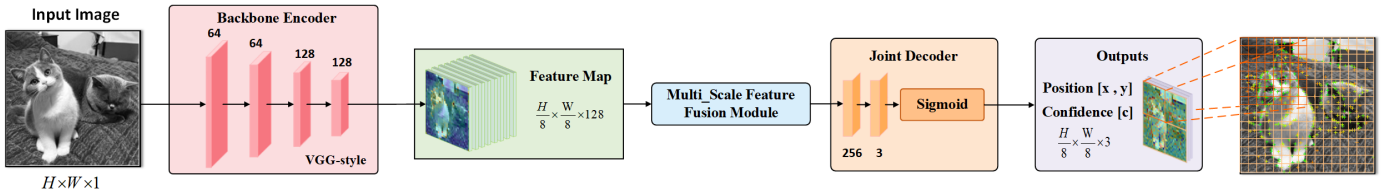


Figure 2. OFPoint pipeline. OFPoint inputs a grayscale image and outputs the keypoint locations and confidences. The multi-scale feature fusion module consists of several multi-scale feature fusion blocks. A single pixel in the output feature map represents 64 pixels in the input grayscale image.

3.1. Backbone Encoder

Since OFPoint requires keypoint positions and confidences for optical flow tracking, we use the visual geometry group (VGG)-style [39] shared encoder from SiLK based on the "joint detection and description" framework as the pre-trained backbone. Fine-tuning enhances the encoder's adaptability for optical flow tracking while maintaining original performance. The backbone encoder includes convolutional layers with a kernel size of 3, max-pooling layers with a stride of 2, rectified linear unit (ReLU) non-linear activation functions, and BatchNorm normalization. After three max-pooling layers, the input grayscale image of size $H \times W \times 1$ is transformed into a dense feature map of output size $\frac{H}{8} \times \frac{W}{8} \times 128$. The dense feature map serves as the input to the multi-scale feature fusion module.

3.2. Multi-Scale Feature Fusion Module

To avoid keypoint tracking based on grayscale patches falling into local minima, we propose a multi-scale feature fusion module composed of several multi-scale feature fusion blocks. The multi-scale feature fusion blocks consist of multi-scale convolutions with channel attention [40] and residual connections [41], modeling the interdependencies between scales, as illustrated in Figure 3. Channel attention, including squeeze and excitation, applies Global Average Pooling to generate a channel descriptor, which is then fed into fully connected layers with ReLU and Sigmoid activations to compute channel-wise weights. The original feature map is recalibrated by multiplying it with these weights, enhancing important feature channels. Residual connections add the input feature map to the output of a series of convolutional layers, mitigating the vanishing gradient problem, which helps preserve low-level features and improves model performance.

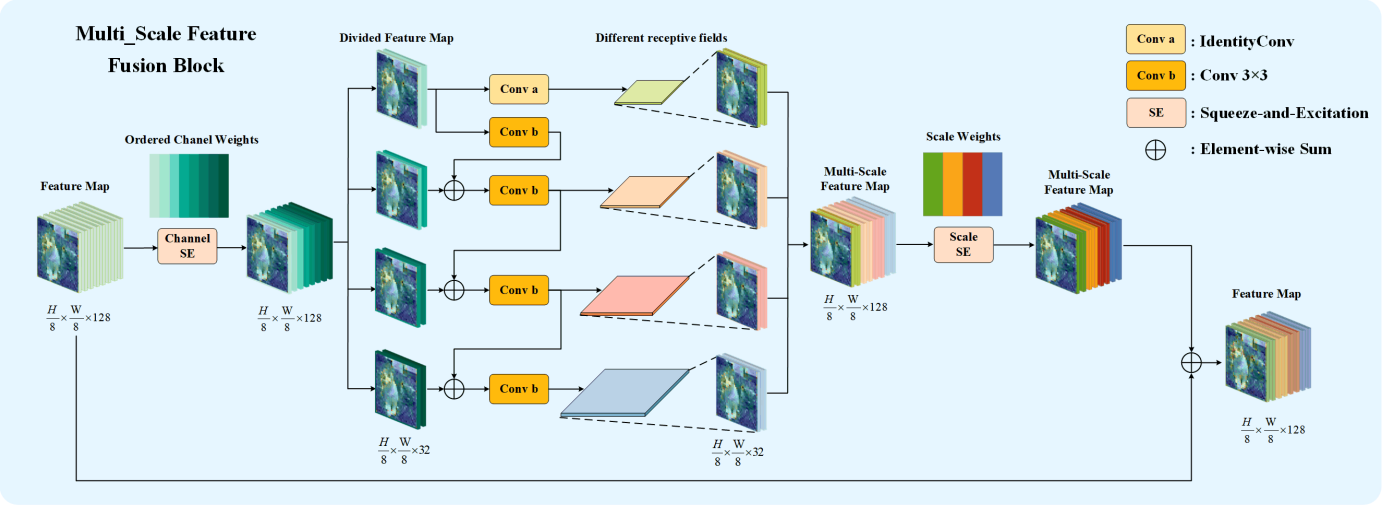


Figure 3. The multi-scale feature fusion module consists of several multi-scale feature fusion blocks, which are composed of multi-scale convolutions, channel attention, and residual connections.

The multi-scale feature fusion block takes dense feature maps from the encoder and generates new maps of the same size. The input-dense feature maps are grouped by channels, and different scale convolution operations are applied to each group. These operations increase the effective receptive field of the feature map by using a different number of convolution kernels [42], effectively simulating convolutions at multiple scales.

Since some feature maps need to undergo multi-scale extraction, we assign them greater significance than the others. To achieve this, we apply channel attention to compute the weight for each channel in the input-dense feature maps and rank the feature maps

based on the channel weights, called Ordered Channel Weights. This ensures that feature maps with higher weights undergo feature extraction at multiple scales.

After the multi-scale feature extraction, we compute the scale attention weights based on channels again. However, instead of considering individual channels, we now consider the average weight of each group based on the previous grouping, called scale weights. Since different groups represent different scales, scale-based attention allows us to adjust the significance of scales in the global feature description, highlighting scale-specific features.

In addition, through residual connections, we add the input feature map to the feature map obtained through multi-scale feature fusion before ReLU, which enhances feature transmission and reuse, thereby improving the model's generalization ability.

The multi-scale feature fusion module consists of two multi-scale feature fusion blocks, with both the input and output feature maps of each block having 128 channels. The input feature map is divided into four groups. The selection of the network structure parameters, specifically the two blocks with 128 channels, is detailed in Section 5.2.4.

3.3. Joint Decoder

In the joint decoder, we optimize the positions and confidences simultaneously to learn repeatable keypoints without explicit definitions. The decoder consists of convolutional layers with a kernel size of 1, followed by a Sigmoid function to constrain the relative positions (x, y) and confidences within the range $[0, 1]$.

Compared to the $H \times W \times 1$ input grayscale image of OFPoint, the joint decoder takes a dense feature map with reduced spatial dimensions and increased channel depth as input and outputs a $\frac{H}{8} \times \frac{W}{8} \times 3$ feature map. To obtain the keypoints and corresponding confidences, post-processing of the output feature map is required. Since the output feature map exhibits a 1:64 scale ratio relative to the input grayscale image of OFPoint, we define each pixel in the output feature map as a cell [43], with each cell containing information fused from $8 \times 8 = 64$ pixels of the input grayscale image.

The output of the Sigmoid function, $[x, y, confidence]$, falls within the range of $[0, 1]$ and is defined relative to the corresponding cell. Within each cell, we select the pixel location with the highest probability from the 64 candidate pixels as the keypoint. The selected pixel's location relative to the cell corresponds to the (x, y) relative coordinates, and its probability represents the confidence. By considering the relative position of the cell in the input grayscale image, we can determine the absolute coordinates of the keypoint in the input grayscale image. Therefore, for a $H \times W \times 1$ grayscale image, it is theoretically possible to detect $\frac{H}{8} \times \frac{W}{8}$ keypoints.

4. Training

Similar to the detector in the "joint detection and description" framework, we employ a self-supervised framework to train OFPoint, determining pixel-level correspondences on Microsoft Common Objects in Context (MS-COCO) 2014 [44]. Image pairs are generated through random homographies, yielding keypoints detected by OFPoint under various viewpoints and illuminations. To obtain pixel-level correspondences, we leverage known homographies to align keypoints in image pairs as valid point pairs, i.e., points can be repeatedly detected or have correspondences without manual definition. We utilize valid point pairs as self-supervised information for training and construct a loss function between image pairs, as illustrated in Figure 4.

Given image pair A and B based on homography H, the distance matrix D is defined by the Euclidean pixel distances between the keypoint matrices P_a and P_b of the image pair. An element d_{ij} in matrix D represents the Euclidean distance between point $p_i^{A \to B} = H p_i^A$

indexed i in Image A and point p_j^B indexed j in Image B, where p_i^A and p_j^B are defined as point pairs. The distance matrix D between point pairs is defined as

$$\mathrm{D} = \left[\mathrm{d}_{ij}\right]_{P^A \times P^B} = \left[\left\| p_i^{A \to B} - p_j^B \right\|_2\right]_{P^A \times P^B} \tag{1}$$

If the distance d_{ij} between point pair (p_i^A, p_j^B) is less than a threshold $\varepsilon = 5$ pixels, and d_{ij} is the shortest distance from point $p_i^{A \to B} = H p_i^A$ indexed i in Image A to all points in Image B, we define the point pair (p_i^A, p_j^B) as a valid point pair.

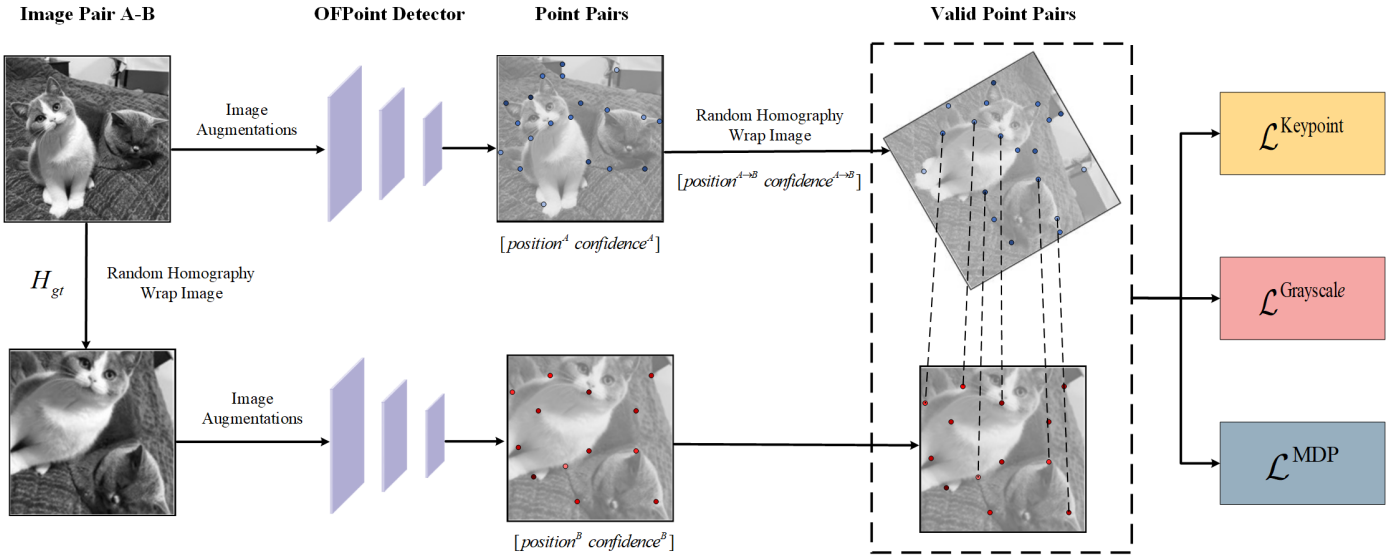


Figure 4. Constructing image pairs and training OFPoint through random homography, including rotation, scaling, and perspective transformation.

4.1. Loss Function

OFPoint's loss function $\mathcal{L}^{\text{Total}}$ comprises keypoint loss term $\mathcal{L}^{\text{Keypoint}}$, grayscale loss term $\mathcal{L}^{\text{Grayscale}}$, and maximum discriminative probability (MDP) loss term $\mathcal{L}^{\text{MDP}}$, all based on valid point pairs. Each loss term is weighted by factors $\alpha_{Keypoint} = 1.0$, $\alpha_{Grayscale} = 1.0$, and $\alpha_{MDP} = 2.0$. The total loss is as follows:

$$\mathcal{L}^{\text{Total}} = \alpha_{\text{Keypoint}} \mathcal{L}^{\text{Keypoint}} + \alpha_{\text{Grayscale}} \mathcal{L}^{\text{Grayscale}} + \alpha_{\text{MDP}} \mathcal{L}^{\text{MDP}} \tag{2}$$

4.1.1. Keypoint Loss Term

The keypoint loss term $\mathcal{L}^{\text{Keypoint}}$ enhances keypoint repeatability and consists of distance loss l_n^{dis}, confidence loss l_n^{conf}, and distance-confidence association loss $l_n^{\text{dis-conf}}$ for N valid point pairs, weighted by factors $\alpha_{dis} = 1.0$, $\alpha_{conf} = 4.0$ and $\alpha_{\text{dis-conf}} = 2.0$.

$$\mathcal{L}^{\text{Keypoint}} = \frac{1}{N}\left(\alpha_{\text{dis}} \sum_{n=1}^{N} l_n^{\text{dis}} + \alpha_{\text{conf}} \sum_{n=1}^{N} l_n^{\text{conf}} + \alpha_{\text{dis-conf}} \sum_{n=1}^{N} l_n^{\text{dis-conf}}\right) \tag{3}$$

As the training of OFPoint progresses, minimizing the Euclidean pixel distance loss l_n^{dis} ensures that valid point pairs correspond to the same point in the space.

$$l_n^{\text{dis}} = \left\| H p_n^A - p_n^B \right\|_2 = \left\| p_n^{A \to B} - p_n^B \right\|_2 \tag{4}$$

The confidence loss l_n^{conf} ensures that the confidences of valid point pairs remain consistent. It is implemented using the L2 norm of the confidence difference between valid point pairs.

$$l_n^{\text{conf}} = \left\| c_n^A - c_n^B \right\|_2 \tag{5}$$

The closer valid point pairs have consistent and higher confidence. To establish a connection between distance and confidence [43], we introduce association loss $l_n^{\text{dis-conf}}$, as shown in Equation (6), where $\bar{d}$ is derived from the distance matrix D.

$$l_n^{\text{dis_conf}} = \left(d_n - \bar{d}\right)\left(c_n^A + c_n^B\right) \tag{6}$$

4.1.2. Grayscale Loss Term

The grayscale loss term $\mathcal{L}^{\text{Grayscale}}$ enhances cosine similarity between grayscale patches of valid point pairs to improve keypoint performance in optical flow tracking. Grayscale patches are centered on valid point pairs, and the grayscale information from the 5×5 region is treated as a grayscale vector. To ensure rotation invariance, the pixel regions are rotated to align with respective centroid orientations using Equation (7), where (x, y) denotes the pixel coordinates relative to the grayscale patches, $I(x, y)$ represents the grayscale value at the corresponding pixel coordinates, and θ denotes the orientation of the grayscale vector.

$$\begin{gathered} \begin{bmatrix} m_{10} \\ m_{01} \end{bmatrix} = \sum_{x=-L}^{L} \sum_{y=-L}^{L} \begin{bmatrix} x \\ y \end{bmatrix} I(x, y) \\ \theta = \arctan 2(m_{01}, m_{10}) \end{gathered} \tag{7}$$

To reduce the illumination effects between grayscale patches, we utilize zero-mean grayscale vectors for cosine similarity, where A and B denote the grayscale vectors corresponding to the grayscale patches, and Rot indicates that the grayscale patches have undergone rotation.

$$\mathcal{L}^{\text{Grayscale}} = 1 - \frac{1}{N}\sum_{n=1}^{N} cossim_n = 1 - \frac{1}{N}\sum_{n=1}^{N} \frac{(A_{Rot}^n - \bar{A}_{Rot}^n) \cdot (B_{Rot}^n - \bar{B}_{Rot}^n)}{||A_{Rot}^n - \bar{A}_{Rot}^n||_2 \cdot ||B_{Rot}^n - \bar{B}_{Rot}^n||_2} \tag{8}$$

4.1.3. Maximum Discriminative Probability Loss Term

Inspired by descriptor constraints, the MDP loss term $\mathcal{L}^{\text{MDP}}$ enforces corresponding constraints between point pairs. It consists of the distance MDP loss $l_n^{\text{mdp-dis}}$ and the grayscale MDP loss $l^{\text{mdp-gray}}$, weighted by factors $\alpha_{\text{mdp-dis}} = 1.0$ and $\alpha_{\text{mdp-gray}} = 0.5$. The MDP, based on softmax probability with the temperature parameter $\tau = 1.5$, identifies distinctive valid point pairs by distance and grayscale similarity. The distance MDP loss $l_n^{\text{mdp-dis}}$ ensures uniform distribution of valid point pairs, while the grayscale MDP loss $l^{\text{mdp-gray}}$ enhances the grayscale similarity between valid pairs relative to non-valid ones.

$$\mathcal{L}^{\text{MDP}} = \alpha_{\text{mdp-dis}} \frac{1}{N}\sum_{n=1}^{N} l_n^{\text{mdp-dis}} + \alpha_{\text{mdp-gray}} l^{\text{mdp-gray}} \tag{9}$$

The distance MDP loss $l_n^{\text{mdp-dis}}$, Equation (10), constitutes the negative log-likelihood of the bidirectional distance probabilities from the distance matrix D, wherein the distinctiveness is embodied in the keypoints level.

$$l_n^{\text{mdp-dis}} = -\log\left(Prob_{i\to j} Prob_{j\to i}\right) = -\left[\log\left(Prob_{i\to j}\right) + \log\left(Prob_{j\to i}\right)\right] \tag{10}$$

The forward distance probability $Prob_{i\to j}$ is computed based on the distances between point pairs and processed via softmax. The normalization term is the sum of distances between point p_i^A indexed i in Image A and all keypoints in Image B. Conversely, the normalization term in the backward distance probability $Prob_{j\to i}$ is the sum of distances between point p_j^B indexed j in Image B and all keypoints in Image A.

$$Prob_{i\to j} = 1 - \frac{e^{\frac{d_{ij}}{\tau}}}{\sum_{n=1}^{N} e^{\frac{d_{in}}{\tau}}} \quad Prob_{j\to i} = 1 - \frac{e^{\frac{d_{ij}}{\tau}}}{\sum_{n=1}^{N} e^{\frac{d_{nj}}{\tau}}} \tag{11}$$

If the pair (p_i^A, p_j^B) represents a unique point pair, the significant softmax probability is achieved. The distance probability $Prob_{i\to j}$ and $Prob_{j\to i}$ are illustrated in Figure 5.

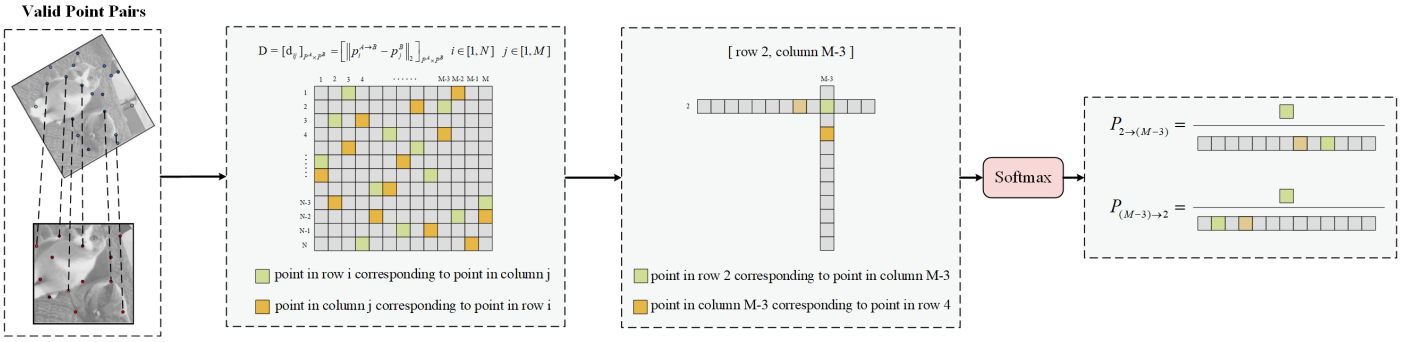


Figure 5. We obtain the maximum discriminative keypoint distance loss through Softmax and utilize the temperature parameter $\tau = 1.5$ to smooth the distance differences between keypoints.

Similar to the keypoint distance MDP loss $l_n^{\text{mdp-dis}}$, the image grayscale MDP loss $l^{\text{mdp-gray}}$ enforces corresponding constraints at the homography estimation level between image pairs. By averaging the cosine similarity of patches from valid point pairs, we obtain the cosine similarity of image pairs.

$$Cos_sim = \frac{1}{N}\sum_{n=1}^{N} cossim_n \tag{12}$$

Assuming higher cosine similarity between grayscale patches of valid point pairs, we hypothesize a correlation between the cosine similarity of image pairs and homography. We randomly shuffle the correspondences among point pairs to compute homographies for different correspondences in Figure 6. If the correspondences are unique and accurate, the estimated homography will closely match the ground truth homography H_{gt}, and the cosine similarity of image pairs will achieve a significant softmax probability compared to others.

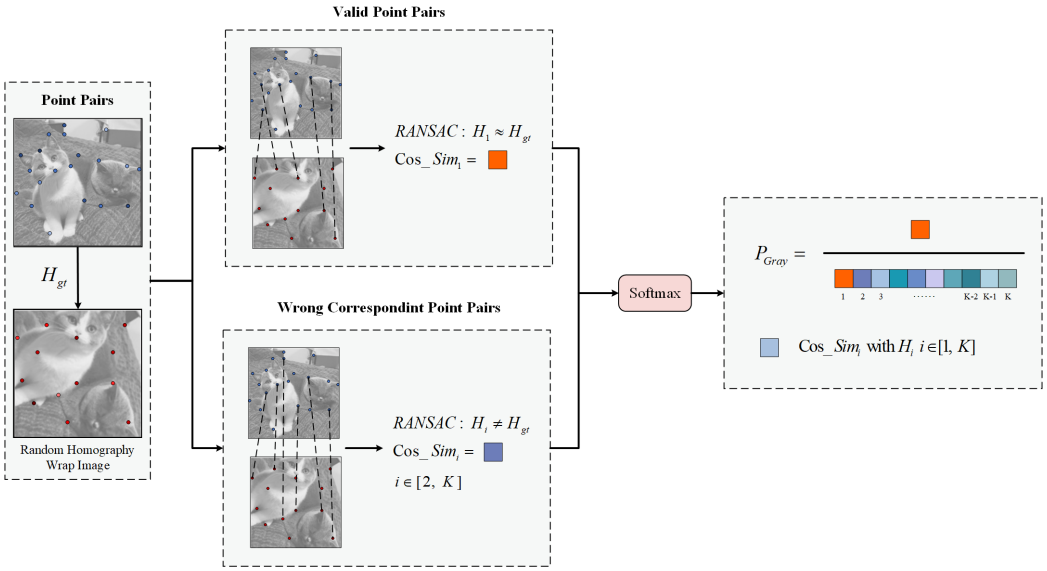


Figure 6. We obtain the maximum discriminative image grayscale loss through Softmax, and utilize the temperature parameter $\tau = 1.5$ to smooth the similarity differences between image grayscale.

Similarly, by leveraging the negative log-likelihood function, we can achieve the image grayscale MDP loss $l^{\text{mdp-gray}}$:

$$l^{\text{mdp-gray}} = -\log\left(Prob_{\text{Gray}}\right) = -\log\left(\frac{e^{\frac{Cos_sim}{\tau}}}{\sum_{k=1}^{K} e^{\frac{Cos_sim_k}{\tau}}}\right) \tag{13}$$

5. Results and Discussion

5.1. Experimental Details

To ensure effective training of OFPoint, it was conducted on an NVIDIA (Santa Clara, CA, USA) 4090 GPU with a batch size of 64, taking approximately 18 h to complete 20 epochs. The Adam optimizer [45] was used with a learning rate set to 0.001. The learning rate was decayed by a factor of 0.1 at 60% and 80% of the training progress. After training OFPoint, we evaluated its performance and integration with VO against representative baselines and SOTA methods.

We assessed the repeatability (Rep) [46] and localization error (LE) [16] in detection metrics; the mean corresponding accuracy (MCA) [24–26] and mean normalized cross-correlation (MNCC) [47] in corresponding metrics; homography estimation accuracy (HEA) [16], and homography estimation AUC (HEAUC) [48,49] in homography estimation metrics (HE Metrics); and frames per second (FPS), giga floating point operations per second (GFLOPs), and parameters per million (Params/M) in runtime metrics [28]. The definitions of the metrics are provided in Sections 5.2.1–5.2.5.

We evaluated the relevant metrics on HPatches [50], where each scene comprises one reference image and five target images with known homography. To simulate optical flow tracking in image sequences, we augmented the test dataset by generating image pairs with small homographies. We compared OFPoint with detectors SIFT, ORB, and GoodFeaturesToTrack (GFTT) [51] using OpenCV interfaces. For SuperPoint, DISK, ALIKE, and SiLK, we used the GitHub versions provided by the authors. To minimize the impact of correspondence establishment, we used both descriptor matching (BruteMatch and Flannmatch) and optical flow tracking (LK sparse optical flow) for evaluation.

To evaluate the VO performance with OFPoint, we utilized a Python-based VO to generate trajectories for comparison with ground truth. We focused on the mean distance error (MDE), relative distance error (RDE), and FPS, conducting the experiment using KITTI's VO/SLAM Evaluation 2012. The VO using OFPoint established correspondences via optical flow tracking, while other VOs used detectors like ORB, SIFT, GFTT, SuperPoint, and SiLK. Correspondence establishment for all VOs was conducted with descriptor matching and optical flow tracking, and pose estimation was performed using Random Sample Consensus (RANSAC) [52].

5.2. Evaluation of OFPoint

5.2.1. Detection Metrics

The detection metrics include repeatability (Rep) and localization error (LE). We established correspondence with ground truth homography H_{gt}, defining a point pair as valid if their distance was less than the threshold $\varepsilon = 3$ pixels. Figure 7 visualizes the detected keypoints across different views of the same scene.

Rep denotes the average ratio of valid points to all keypoints in two views, while LE denotes the average pixel distance between valid point pairs. Table 1 shows that OFPoint outperforms other detectors with higher Rep and lower LE, achieving performance comparable to SiLK and detecting feature points across different images.

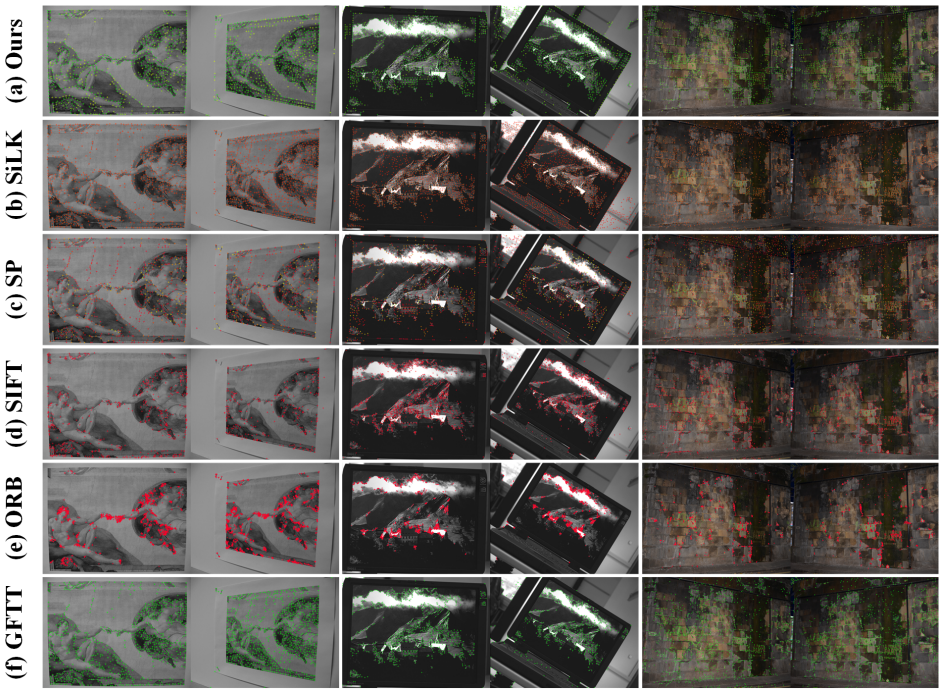


Figure 7. (**a**) OFPoint, (**b**) SiLK, (**c**) SuperPoint, (**d**) SIFT, (**e**) ORB, and (**f**) GFTT. The keypoint distributions in (**a**–**c**) are uniform, allowing for the detection of more details on the image.

Table 1. Detection metrics for detectors on HPatches.

Detectors	Size	Rep ↑	LE (Pixel) ↓
GFTT	240 × 320	0.463	**0.880**
	480 × 640	0.425	**0.894**
ORB	240 × 320	0.538	1.132
	480 × 640	0.523	1.221
SIFT	240 × 320	0.507	0.888
	480 × 640	0.502	1.024
SP	240 × 320	0.639	1.050
	480 × 640	0.611	1.141
DISK	240 × 320	0.634	0.892
	480 × 640	0.587	1.031
ALIKE	240 × 320	0.627	0.955
	480 × 640	0.579	1.125
SiLK	240 × 320	**0.676**	0.899
	480 × 640	**0.652**	1.091
Ours	240 × 320	**0.659**	**0.993**
	480 × 640	**0.632**	**1.076**

Note: Arrows indicate desired direction (↑: higher better, ↓: lower preferred). Bold values highlight the metrics for OFPoint and the best-performing metrics for other detectors.

5.2.2. Corresponding Metrics

The corresponding metrics include mean corresponding accuracy (MCA) and mean normalized cross-correlation (MNCC). MCA is the average ratio of valid point pairs to corresponding point pairs, while MNCC measures the similarity between patches of valid point pairs for optical flow tracking. Figure 8 shows the keypoint correspondences across different views.

Without the ground truth homography H_{gt}, we employed both descriptor matching and optical flow tracking to establish keypoint correspondences between two views. Since OFPoint does not generate keypoint descriptors, OpenCV's SIFT descriptor interface was employed to extract descriptors for the keypoints detected by OFPoint, thereby facilitating

keypoint matching. By correspondence establishment, the corresponding point pairs were obtained. Valid point pairs are defined as those with a reprojection distance less than the threshold $\varepsilon = 3$ pixels. Table 2 shows the correspondence results with different methods for establishing correspondence.

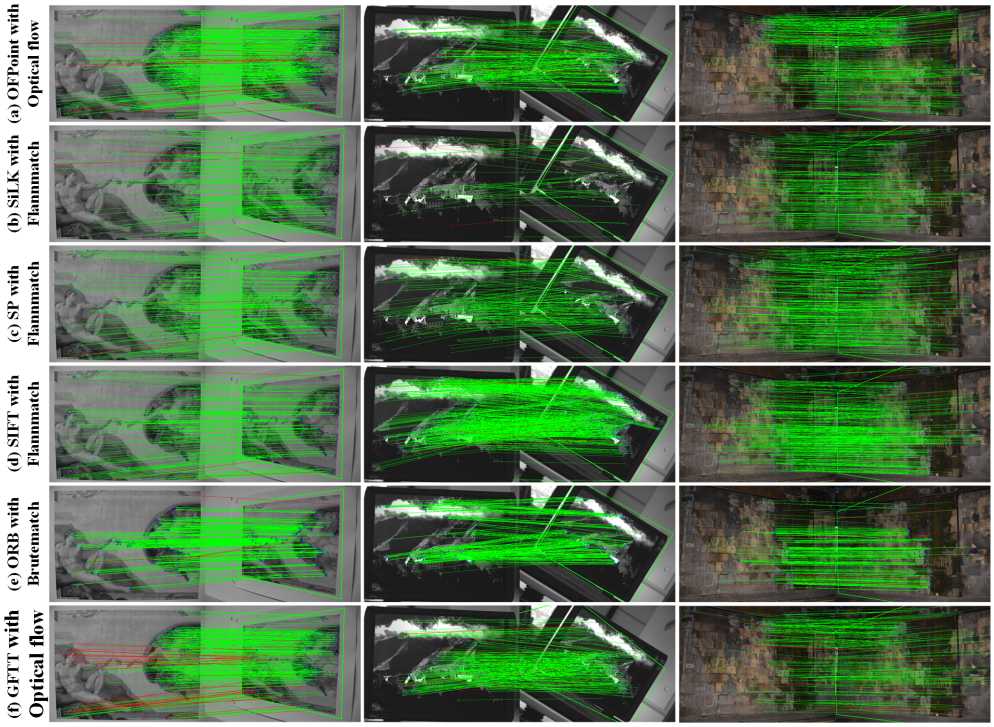


Figure 8. The green bounding box represents the homography estimation. Optical flow tracking based on OFPoint can achieve a homography estimation comparable to that of descriptor matching.

Table 2. Corresponding metrics for detectors on HPatches.

Detectors	Size	Descriptor Matching		Optical Flow Tracking	
		MCA ↑	MNCC ↑	MCA ↑	MNCC ↑
GFTT	240 × 320	0.317	0.504	0.459	0.537
	480 × 640	0.303	0.506	0.402	0.482
ORB	240 × 320	0.314	0.288	0.468	**0.554**
	480 × 640	0.309	0.301	0.426	0.493
SIFT	240 × 320	0.489	0.418	0.452	0.486
	480 × 640	0.457	0.427	0.420	0.425
SP	240 × 320	0.689	0.485	0.464	0.533
	480 × 640	**0.656**	0.502	0.421	0.467
DISK	240 × 320	**0.692**	0.567	0.476	0.445
	480 × 640	0.643	0.572	0.413	0.397
ALIKE	240 × 320	0.642	**0.605**	**0.479**	0.450
	480 × 640	0.612	**0.619**	0.421	0.419
SiLK	240 × 320	0.653	0.556	0.477	0.551
	480 × 640	0.629	0.572	**0.443**	**0.496**
Ours	240 × 320	**0.639**	**0.659**	**0.517**	**0.584**
	480 × 640	**0.617**	**0.678**	**0.462**	**0.523**

Note: Arrows indicate desired direction (↑: higher better). Bold values highlight the metrics for OFPoint and the best-performing metrics for other detectors.

The keypoints detected by OFPoint established correspondence using either descriptor matching, achieving performance comparable to other detectors on MCA, or optical flow tracking, which exhibits significant improvements on MCA. Additionally, OFPoint achieves

the highest MNCC, demonstrating superior grayscale consistency and making its keypoints more suitable for optical flow tracking.

5.2.3. Homography Estimation Metrics

The homogeneity metrics include homography estimation accuracy (HEA) and homography estimation AUC (HEAUC). We estimated homography between two views using RANSAC based on valid point pairs and established keypoint correspondences via descriptor matching and optical flow tracking, with details in Section 5.2.2.

The homography estimation error is the average distance error between the four corner points of the image transformed by homography H_{gt} and the estimated homography H_{est}, with valid estimated homographies determined by thresholds $\varepsilon = 1, 3, 5$ pixels. HEA is the ratio of valid estimated homographies, while HEAUC calculates the area under the curve between homography estimation error and recall rate.

Table 3 shows that, with descriptor matching, OFPoint keypoints show a slight deficit compared to SiLK in terms of HEA and HEAUC, but they outperform other detectors. However, with optical flow tracking, OFPoint keypoints surpass others in both HEA and HEAUC.

Table 3. Estimated Homographies for detectors on HPatches.

Detectors	Size	Descriptor Matching						Optical Flow Tracking					
		HEA ↑			HEAUC ↑			HEA ↑			HEAUC ↑		
		$\varepsilon = 1$	$\varepsilon = 3$	$\varepsilon = 5$	$\varepsilon = 1$	$\varepsilon = 3$	$\varepsilon = 5$	$\varepsilon = 1$	$\varepsilon = 3$	$\varepsilon = 5$	$\varepsilon = 1$	$\varepsilon = 3$	$\varepsilon = 5$
GFTT	240 × 320	0.095	0.272	0.353	0.040	0.145	0.213	0.252	0.395	0.434	0.117	0.268	0.328
	480 × 640	0.151	0.382	0.450	0.071	0.217	0.297	0.202	0.386	0.445	0.089	0.239	0.311
ORB	240 × 320	0.116	0.419	0.531	0.039	0.204	0.316	0.145	0.362	0.421	0.058	0.206	0.282
	480 × 640	0.229	0.522	0.620	0.102	0.310	0.418	0.193	0.388	0.471	0.079	0.239	0.318
SIFT	240 × 320	0.589	0.776	0.815	0.293	**0.577**	0.665	0.284	0.483	0.534	0.129	0.320	0.396
	480 × 640	**0.529**	0.722	0.775	**0.294**	0.537	0.623	0.293	0.456	0.515	0.131	0.319	0.391
SP	240 × 320	0.527	**0.856**	0.910	0.219	0.576	**0.701**	0.310	**0.501**	0.553	0.145	0.335	0.415
	480 × 640	0.489	**0.822**	**0.883**	0.229	0.543	0.668	0.284	0.480	**0.524**	0.141	0.319	**0.393**
DISK	240 × 320	0.502	0.829	0.898	0.240	0.562	0.685	0.281	0.462	0.515	0.124	0.304	0.380
	480 × 640	0.403	0.779	0.872	0.184	0.491	0.631	0.258	0.443	0.498	0.127	0.286	0.367
ALIKE	240 × 320	0.503	0.841	0.891	0.242	0.571	0.691	**0.315**	0.500	**0.572**	**0.149**	0.337	**0.417**
	480 × 640	0.456	0.755	0.817	0.231	0.512	0.624	**0.300**	0.475	0.508	0.135	**0.322**	0.391
SiLK	240 × 320	**0.612**	0.855	**0.928**	**0.346**	0.573	0.692	0.309	0.498	0.570	0.136	**0.345**	0.391
	480 × 640	0.525	0.766	0.879	0.272	**0.564**	**0.674**	0.271	**0.483**	0.521	**0.144**	0.296	0.388
Ours	240 × 320	**0.586**	**0.842**	**0.905**	**0.286**	**0.581**	**0.689**	**0.322**	**0.506**	**0.583**	**0.139**	**0.356**	**0.412**
	480 × 640	**0.514**	**0.776**	**0.854**	**0.275**	**0.555**	**0.657**	**0.294**	**0.485**	**0.542**	**0.148**	**0.338**	**0.407**

Note: Arrows indicate desired direction (↑: higher better). Bold values highlight the metrics for OFPoint and the best-performing metrics for other detectors.

5.2.4. Ablation Study of OFPoint

To evaluate the performance of various modules in OFPoint, We evaluated different configurations on 240 × 320 resolution images. The baseline employed an initialized VGG as the backbone encoder with a loss term for keypoint location and confidence. Model 1 replaced the initialized VGG with a pre-trained VGG. Model 2 introduced the multi-scale feature fusion module to the baseline. Model 3 incorporated the novel loss term. Finally, OFPoint combined all modules together. For keypoint correspondence, we chose optical flow tracking.

In Table 4, the metrics for baseline and OFPoint are presented at the top and bottom. The middle rows show the improvement ratios of Models 1–3 relative to the baseline. We evaluated the detection metrics, corresponding metrics, and homography estimation metrics (HE Metrics), with the definitions in Section 5.2.1–5.2.3. The ablation results demonstrate that the pre-trained model enhances both detection metrics and HE Metrics.

Additionally, the novel loss term significantly affects MNCC. Compared to the baseline, OFPoint exhibits consistent improvements across various metrics

Table 4. The influence of various configurations of OFPoint on the metrics.

Models	Backbone Encoder	Multi-Scale	Novel Loss	Detection Metrics		Corresponding Metrics		HE Metrics	
				Rep ↑	LE (Pixel) ↓	MCA ↑	MNCC ↑	HEA ↑	HEAUC ↑
Baseline				0.632	1.538	0.481	0.514	0.431	0.297
Model 1	○			2.8%	**20.6**%	4.7%	5.6%	**10.8**%	**12.3**%
Model 2		○		**3.1%**	16.6%	2.6%	0.7%	4.3%	4.0%
Model 3			○	0.5%	4.2%	**5.1**%	**10.4**%	8.2%	9.7%
OFPoint	○	○	○	0.659	0.993	0.517	0.584	0.506	0.356

Note: Arrows indicate desired direction (↑: higher better, ↓: lower preferred). The symbol ○ represents the different modules of the model. Bold values highlight the best-performing metrics for Ablation Study.

In addition, we considered the impact of OFPoint architecture parameters on keypoint detection. Since the shared encoder and joint decoder are fixed, we aimed to improve keypoint detection by comparing different parameters, namely, the number of multi-scale feature fusion blocks and the number of feature map channels in the multi-scale feature fusion module. The optimal parameters were identified based on the details of keypoint detection and the model size.

For the number of blocks and channels in the multi-scale feature fusion module, we compared four parameter configurations: Model A employed 2 blocks with 128 channels, Model B employed 4 blocks with 128 channels, Model C employed 2 blocks with 256 channels, and Model D employed 4 blocks with 256 channels. Since the differences in keypoint detection across these architecture parameters were minor, with variations primarily in fine details, we selected the image with weak-texture environments from KITTI for testing.

As shown in Figure 9, Model A was capable of detecting comprehensive keypoints and capturing finer details, as evidenced by the precise corner details of the windows in the image. Models B and C performed similarly to Model A but exhibited false detections on weak-texture walls and missed some details. Compared to the other models, Model D exacerbated the issue of false keypoint detections, making it unable to detect more objects and details.

Although the results of the four parameter configurations are not significantly different, their model sizes vary considerably, as shown in the Table 5. The parameter size of Model A is 1.278 M, while the maximum parameter size of the other models reaches 3.712 M. Notably, the parameter size of SiLK is only 0.942 M, and excessive model parameters do not meet the requirements for lightweight deployment. Considering both keypoint detection and model complexity, we selected the configuration with 2 blocks and 128 channels as the optimal setting.

Table 5. Model sizes for different parameter configurations.

Model	Blocks	Channels	Params/M ↓
Model A	**2**	**128**	**1.278**
Model B	4	128	1.921
Model C	2	256	2.320
Model D	4	256	3.712
SuperPoint	-	-	1.301
SiLK	-	-	0.942

Note: Arrows indicate desired direction (↓: lower preferred). Bold values represent the number of Blocks and channels, Params/M corresponding to the OFPoint structure.

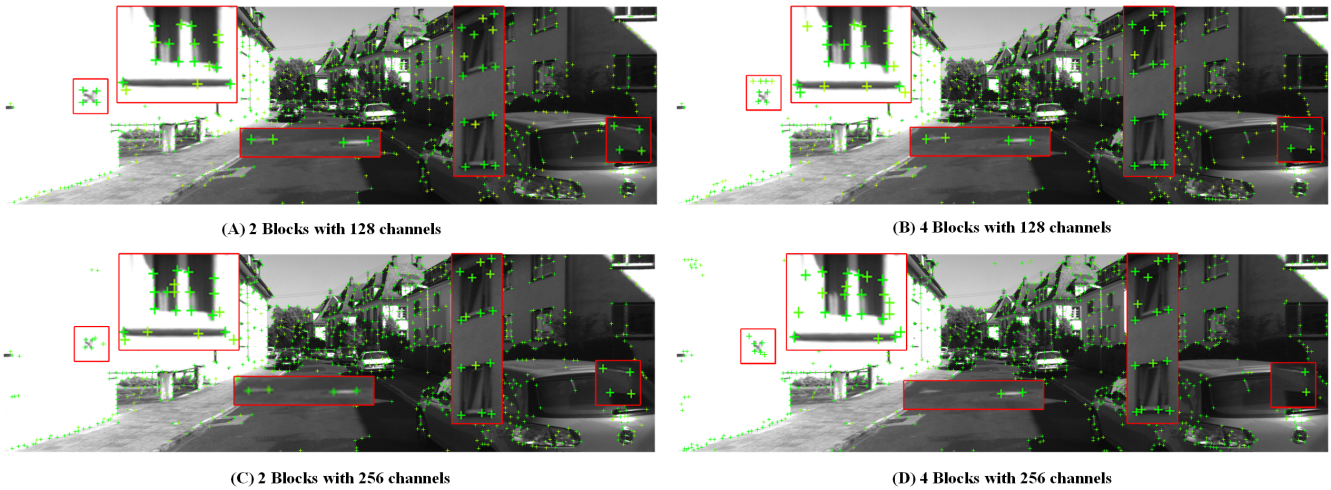


Figure 9. Four different parameter configurations were employed to evaluate the effectiveness of OFPoint in detecting keypoints and capturing details in the scene. The red boxes in the image highlight the details detected by OFPoint on different objects.

5.2.5. Runtime Metrics

To evaluate the feasibility of integrating keypoint detectors into VO, we measured frames per second (FPS), giga floating point operations per second (GFLOPs), and parameters per million (Params/M). Table 6 shows the inference times of the detectors. CNN-based network inferences were evaluated on an NVIDIA 3060 Laptop GPU, while ORB, SIFT, and GFTT were evaluated on an Intel i7-12700HQ CPU. Despite incorporating post-processing steps such as handling output into its inference, OFPoint maintains strong real-time performance due to the elimination of descriptor extraction.

Table 6. Runtime metrics for detectors on HPatches.

Detectors	Size	FPS ↑	GFLOPs ↓	Params/M ↓
GFTT	240 × 320	2108	-	-
	480 × 640	524		
ORB	240 × 320	593	-	-
	480 × 640	200		
SIFT	240 × 320	163	-	-
	480 × 640	31		
SP	240 × 320	167	24.812	1.301
	480 × 640	67	210.900	
DISK	240 × 320	88	24.758	1.092
	480 × 640	26	99.030	
ALIKE	240 × 320	**249**	**2.131**	**0.329**
	480 × 640	**90**	**7.991**	
SiLK	240 × 320	76	65.221	0.942
	480 × 640	22	275.132	
Ours	240 × 320	**254**	**6.896**	**1.278**
	480 × 640	**86**	**27.584**	

Note: Arrows indicate desired direction (↑: higher better, ↓: lower preferred). Bold values highlight the metrics for OFPoint and the best-performing metrics for other detectors.

As shown in Sections 5.2.1–5.2.5, descriptor matching offers higher accuracy in correspondence. However, optical flow tracking, while sacrificing some accuracy, is more suitable for VO due to its speed. OFPoint-detected keypoints, with their grayscale consistency and local stability, improve the accuracy of optical flow tracking, narrowing the gap to descriptor matching.

5.3. Evaluation of VO with OFPoint

The VO performance is evaluated using mean distance error (MDE), relative distance error (RDE), and FPS [11]. MDE is the average Euclidean distance in two dimensions between the estimated values and the ground truth values. RDE is the Euclidean distance between differences in consecutive estimates and ground truths. Unlike Section 5.2.5, FPS reflects the running time of VO, including keypoint detection, correspondence establishment, and homography estimation.

To isolate detector performance in VO, we used descriptor matching and optical flow tracking for correspondence establishment. The estimated trajectories with different VO configurations and ground truth are shown in Figure 10. Regardless of the correspondence method, trajectories from ORB and GFTT deviate significantly from ground truth, while others show minor discrepancies, with MDEs in double or single digits.

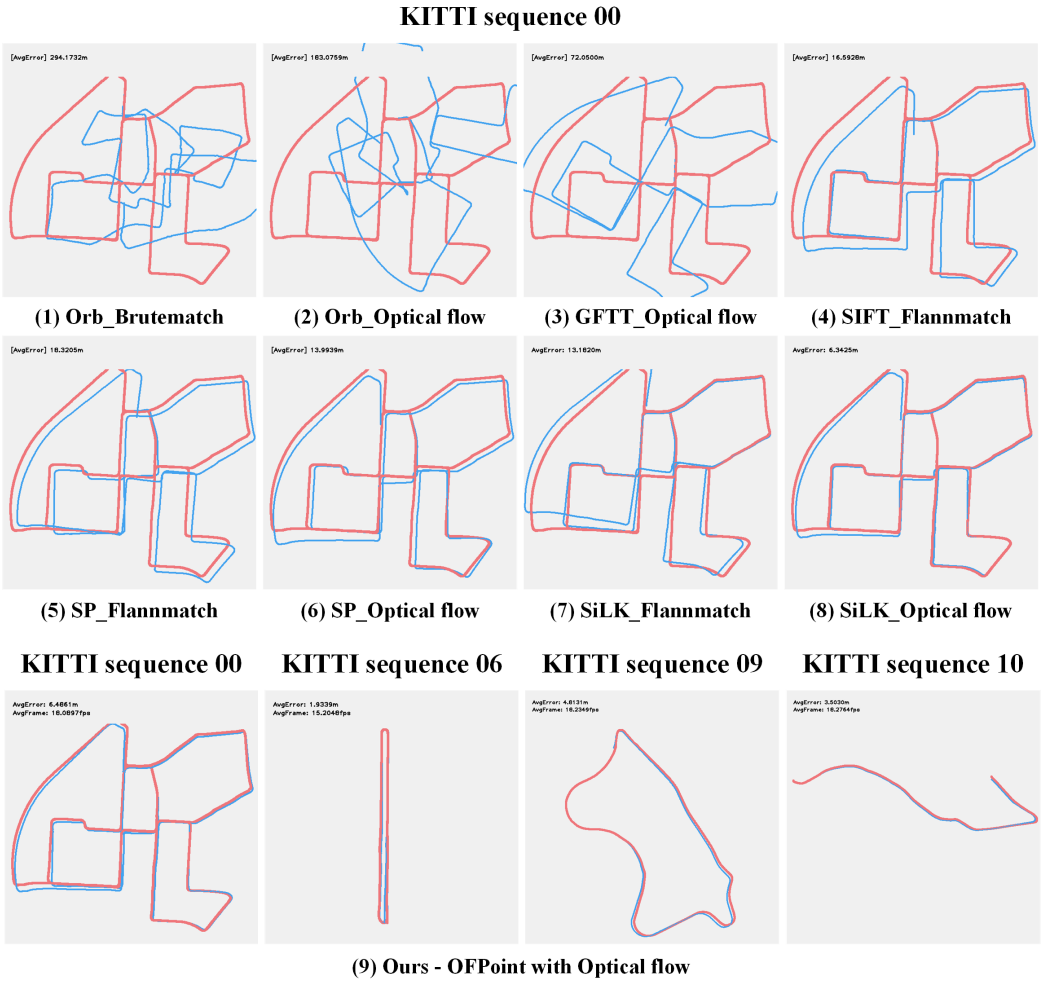


Figure 10. A total of nine different VO configurations, using various keypoint detectors and correspondences establishment methods, are evaluated, including (**1**) Orb with BruteMatch, (**2**) Orb with optical flow, (**3**) GFTT with optical flow, (**4**) SIFT with Flannmatch, (**5**) SuperPoint with Flannmatch, (**6**) SuperPoint with optical flow, (**7**) SiLK with Flannmatch, (**8**) SiLK with optical flow, (**9**) Ours' OFPoint with optical flow. The red curve represents the ground truth, while the blue curve represents the estimated values in two dimensions.

The MDE reflects the overall deviation between the estimated trajectory and the ground truth, while the RDE captures the local error variation between consecutive frames, revealing performance under dynamic changes and system stability over time. Figure 11 shows the MDE and RDE with three dimensions. The MDE in Figure 11 increases more than

in the estimated trajectory, but VO with OFPoint still maintains a smaller MDE. Additionally, VO with OFPoint using optical flow tracking achieves a lower RDE, indicating better local error consistency across frames.

Smaller MDEs, RDEs, and higher FPSs indicate better VO performance. Table 7 shows the evaluation metrics on KITTI sequence 00 (3724.187 m). Since KITTI consists of continuous image frames, optical flow tracking performs well in sequence 00 for the same keypoint detector. Considering the time efficiency of Python 3.8, the C++-based VO should exhibit a higher FPS. The results demonstrate that VO with the optical flow-based OFPoint strikes a better balance between accuracy and real-time performance.

Table 7. Application of VO on KITTI seq 00.

VO	MDE(m) ↓	RDE(m) ↓	FPS ↑
ORB with BruteMatch	294.173	1.470	25
ORB with optical	183.075	0.840	22
GFTT with optical	72.050	0.335	31
SIFT with Flannmatch	16.593	0.124	9
SP with Flannmatch	18.321	**0.079**	14
SP with optical	13.994	0.103	**17**
SiLK with Flannmatch	13.182	0.091	4
SiLK with optical	**6.343**	0.127	7
Ours	**6.486**	**0.065**	**18**

Note: Arrows indicate desired direction (↑: higher better, ↓: lower preferred). Bold values highlight the metrics for OFPoint and the best-performing metrics for other detectors.

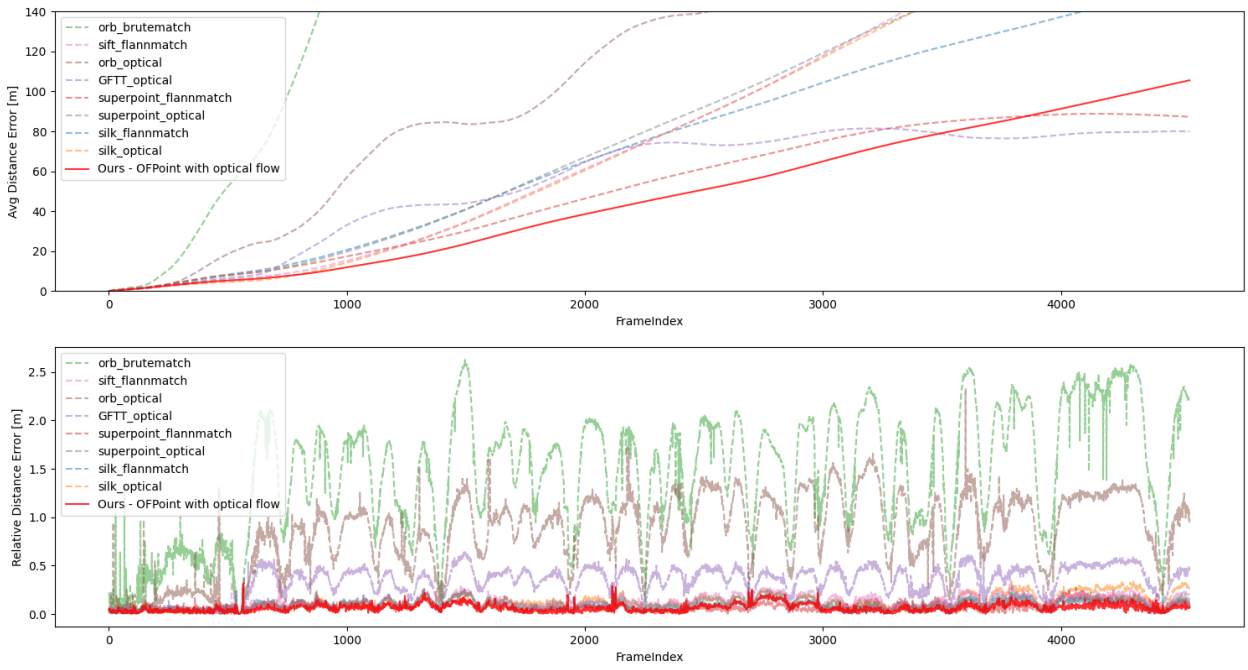


Figure 11. The MDE and RDE in three dimensions (x, y, z) are shown. The red curve, labeled Ours, represents the result obtained from OFPoint using optical flow.

5.4. AR Application Utilizing OFPoint-Based VO

VO provides accurate localization and tracking capabilities for AR, serving as a crucial foundation for the implementation of AR. To evaluate the localization accuracy of the OFPoint-based VO in AR applications, we integrated it into the AR framework, as depicted in Figure 12. The estimated pose from VO, in combination with Unity3D's virtual camera plugin, was employed to align the coordinate systems, cameras, and positions between the

virtual and real worlds, thereby enabling the seamless integration of virtual models with the real-world scene.

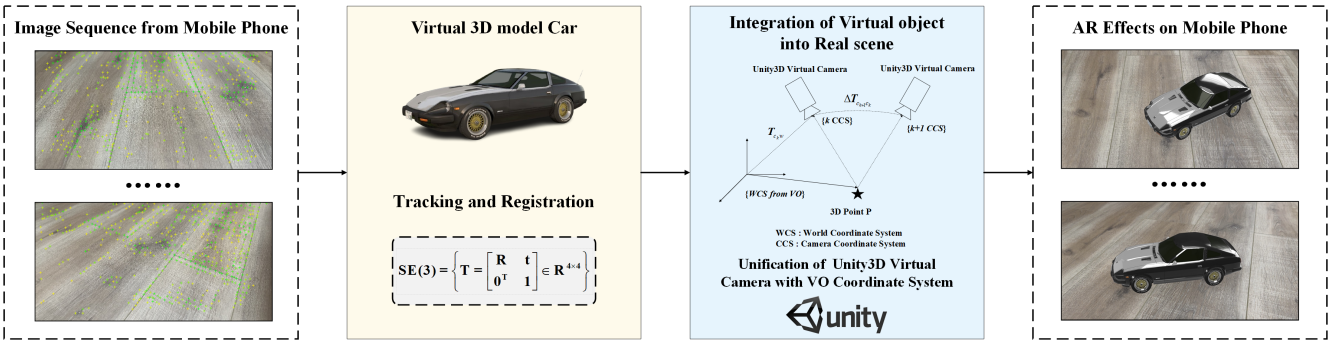


Figure 12. The OFPoint-based VO performs real-time tracking to calculate the pose of the mobile phone in the world coordinate system.

The AR applications used images captured by iPhone 14 via DroidCam 5.0, with 640 × 480 resolution and 30 FPS. The AR application operated in real-time on Ubuntu 20.04, enabling the integration of virtual 3D models into real-world scenes. Figure 13 shows two AR examples, demonstrating the effectiveness of the proposed method in complex scenarios. The AR framework was based on an open-source GitHub project https://github.com/ygx2011/AR_course.git (accessed on 14 February 2025), and the virtual 3D models were sourced from Sketchfab https://sketchfab.com/feed (accessed on 14 February 2025).

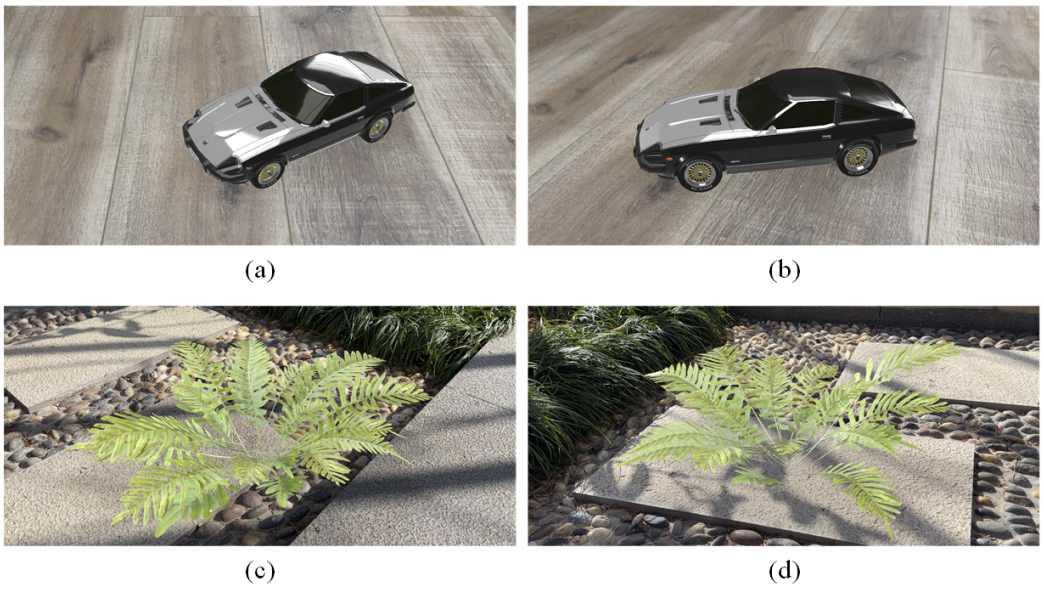


Figure 13. AR effects from the mobile phone camera: (**a**,**b**) car indoors from different perspectives; (**c**,**d**) plant outdoors from different perspectives.

6. Conclusions

This paper presents a VO system based on OFPoint, an end-to-end keypoint detector that eliminates the need for descriptor extraction. OFPoint replaces traditional keypoint detection and establishes correspondences through optical flow tracking. Combining transfer learning, a multi-scale attention mechanism, and MDP loss term, OFPoint jointly outputs keypoint locations and confidences. Inspired by SiLK, OFPoint achieves performance comparable to SOTA on HPatches with the same correspondence method and outperforms other detectors in time performance.

Compared to other methods, VO with OFPoint ensures grayscale consistency and local stability of keypoints during optical flow tracking, improving correspondence accuracy without descriptors. Additionally, it achieves a balance between accuracy and real-time performance, as demonstrated in both KITTI and AR applications. Future work will focus on further enhancing the robustness of OFPoint, creating a lightweight version of OFPoint through model compression [53], and exploring its integration into more complex graphics applications [54].

Author Contributions: Conceptualization, L.S. and Y.W.; methodology, Y.W.; software, Y.W.; validation, L.S., Y.W. and W.Q.; formal analysis, W.Q.; investigation, L.S.; data curation, Y.W.; writing—original draft preparation, Y.W.; writing—review and editing, L.S.; visualization, Y.W.; supervision, W.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Key R&D Program of Jiangsu Province under Grant BE2023010-3, Jiangsu modern agricultural industry single technology research and development project under Grant CX(23)3120, the fund of Laboratory for Advanced Computing and Intelligence Engineering and National Key Laboratory of Ship Structural Safety.

Data Availability Statement: The data utilized in this study are the publicly available HPatches and KITTI datasets. The HPatches dataset is available at https://github.com/hpatches/hpatches-dataset (accessed on 14 February 2025), and the KITTI dataset is available at https://www.cvlibs.net/datasets/kitti/eval_odometry.php (accessed on 14 February 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ALIKE	Accurate and Lightweight Keypoint Detection and Descriptor Extraction
AR	Augmented Reality
CNN	Convolutional Neural Network
DISK	DIScrete Keypoints
D2	Joint Description and Detection
FPS	Frames Per Second
GFLOPs	Giga Floating Point Operations Per Second
GFTT	GoodFeaturesToTrack
HEA	Homography Estimation Accuracy
HEAUC	Homography Estimation Area Under the Curve
HE Metrics	Homography Estimation Metrics
HS	Horn and Schunck
LE	Localization Error
LF	Learning Local Features from Images
LIFT	Learned Invariant Feature Transform
LK	Lucas and Kanade
MCA	Mean Corresponding Accuracy
MDE	Mean Distance Error
MDP	Maximum Discriminative Probability
MNCC	Mean Normalized Cross-Correlation
MS-COCO	Microsoft Common Objects in Context
ORB	Oriented FAST and rotated BRIEF
Params/M	Parameters per Million
PLK	Pyramidal Lucas–Kanade
RAFT	Recurrent All-Pairs Field Transforms
RANSAC	Random Sample Consensus
RDE	Relative Distance Error

ReLU	Rectified Linear Unit
Rep	Repeatability
RoRDs	Rotation-Robust Descriptors
R2d2	Repeatable and Reliable Detector and Descriptor
SIFT	Scale-Invariant Feature Transform
SiLK	Simple Learned Keypoints
SLAM	Simultaneous Localization and Mapping
SOTA	State-of-the-Art
SP	SuperPoint
SPyNet	Spatial Pyramid Network
SURF	Speeded-Up Robust Features
SVO	Semidirect Visual Odometry
TV-L1	Total Variation L1
VGG	Visual Geometry Group
VINS	Visual-Inertial System
VO	Visual Odometry

References

1. Nistér, D.; Naroditsky, O.; Bergen, J. Visual odometry. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, CVPR 2004, Washington, DC, USA, 27 June–2 July 2004; Volume 1, p. I.
2. Mouragnon, E.; Lhuillier, M.; Dhome, M.; Dekeyser, F.; Sayd, P. Real time localization and 3d reconstruction. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; Volume 1; pp. 363–370.
3. Arena, F.; Collotta, M.; Pau, G.; Termine, F. An overview of augmented reality. *Computers* **2022**, *11*, 28. [CrossRef]
4. Ma, J.; Jiang, X.; Fan, A.; Jiang, J.; Yan, J. Image matching from handcrafted to deep features: A survey. *Int. J. Comput. Vis.* **2021**, *129*, 23–79. [CrossRef]
5. Lindeberg, T. Scale invariant feature transform. *Scholarpedia* **2012**, *7*, 10491. [CrossRef]
6. Bay, H.; Tuytelaars, T.; Van Gool, L. Surf: Speeded up robust features. In Proceedings of the Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Proceedings, Part I 9; Springer: Berlin/Heidelberg, Germany, 2006; pp. 404–417.
7. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
8. Kelman, A.; Sofka, M.; Stewart, C.V. Keypoint descriptors for matching across multiple image modalities and non-linear intensity variations. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–7.
9. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [CrossRef]
10. Lucas, B.D.; Kanade, T. An iterative image registration technique with an application to stereo vision. In Proceedings of the IJCAI'81: 7th International Joint Conference on Artificial Intelligence, Vancouver, BC, Canada, 24–28 August 1981; Volume 2, pp. 674–679.
11. Qin, T.; Li, P.; Shen, S. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [CrossRef]
12. Forster, C.; Zhang, Z.; Gassner, M.; Werlberger, M.; Scaramuzza, D. SVO: Semidirect visual odometry for monocular and multicamera systems. *IEEE Trans. Robot.* **2016**, *33*, 249–265. [CrossRef]
13. Jia, K. Image matching algorithm based on grayscale and its improvement. In Proceedings of the 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC), Chengdu, China, 19–21 December 2013; pp. 1203–1207.
14. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
15. Xu, S.; Chen, S.; Xu, R.; Wang, C.; Lu, P.; Guo, L. Local feature matching using deep learning: A survey. *Inf. Fusion* **2024**, *107*, 102344. [CrossRef]
16. DeTone, D.; Malisiewicz, T.; Rabinovich, A. Superpoint: Self-supervised interest point detection and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 224–236.

17. Gleize, P.; Wang, W.; Feiszli, M. SiLK: Simple Learned Keypoints. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 22499–22508.
18. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2009**, *22*, 1345–1359. [CrossRef]
19. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
20. Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the Alvey Vision Conference, Citeseer, Manchester, UK, 18–20 September 1988; Voulme 15, pp. 10–5244.
21. Tareen, S.A.K.; Saleem, Z. A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In Proceedings of the 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Jakarta, Indonesia, 13–14 March 2018; pp. 1–10.
22. Yi, K.M.; Trulls, E.; Lepetit, V.; Fua, P. Lift: Learned invariant feature transform. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part VI 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 467–483.
23. Ono, Y.; Trulls, E.; Fua, P.; Yi, K.M. LF-Net: Learning local features from images. *Adv. Neural Inf. Process. Syst.* **2018**, *31*.
24. Dusmanu, M.; Rocco, I.; Pajdla, T.; Pollefeys, M.; Sivic, J.; Torii, A.; Sattler, T. D2-net: A trainable cnn for joint description and detection of local features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 8092–8101.
25. Revaud, J.; De Souza, C.; Humenberger, M.; Weinzaepfel, P. R2d2: Reliable and repeatable detector and descriptor. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1113–1123.
26. Tyszkiewicz, M.; Fua, P.; Trulls, E. DISK: Learning local features with policy gradient. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 14254–14265.
27. Parihar, U.S.; Gujarathi, A.; Mehta, K.; Tourani, S.; Garg, S.; Milford, M.; Krishna, K.M. Rord: Rotation-robust descriptors and orthographic views for local feature matching. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Online, 27 September–1 October 2021; pp. 1593–1600.
28. Zhao, X.; Wu, X.; Miao, J.; Chen, W.; Chen, P.C.; Li, Z. Alike: Accurate and lightweight keypoint detection and descriptor extraction. *IEEE Trans. Multimedia* **2022**. [CrossRef]
29. Wang, X.; Liu, Z.; Hu, Y.; Xi, W.; Yu, W.; Zou, D. Featurebooster: Boosting feature descriptors with a lightweight neural network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7630–7639.
30. Zhai, M.; Xiang, X.; Lv, N.; Kong, X. Optical flow and scene flow estimation: A survey. *Pattern Recognit.* **2021**, *114*, 107861. [CrossRef]
31. Horn, B.K.; Schunck, B.G. Determining optical flow. *Artif. Intell.* **1981**, *17*, 185–203. [CrossRef]
32. Bouguet, J.Y. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corp.* **2001**, *5*, 4.
33. Papenberg, N.; Bruhn, A.; Brox, T.; Didas, S.; Weickert, J. Highly accurate optic flow computation with theoretically justified warping. *Int. J. Comput. Vis.* **2006**, *67*, 141–158. [CrossRef]
34. Romera, T.; Petreto, A.; Lemaitre, F.; Bouyer, M.; Meunier, Q.; Lacassagne, L.; Etiemble, D. Optical flow algorithms optimized for speed, energy and accuracy on embedded GPUs. *J. Real Time Image Process.* **2023**, *20*, 32. [CrossRef]
35. Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; Brox, T. Flownet: Learning optical flow with convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2758–2766.
36. Ranjan, A.; Black, M.J. Optical flow estimation using a spatial pyramid network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4161–4170.
37. Teed, Z.; Deng, J. Raft: Recurrent all-pairs field transforms for optical flow. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part II 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 402–419.
38. Andriyanov, N.; Dementiev, V.; Tashlinskiy, A. Optimization of the computer vision system for the detection of moving objects. In Proceedings of the International Conference on Pattern Recognition, Montpellier, France, 16–20 January 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 424–431.
39. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* **2014**, abs/1409.1556.
40. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
42. Gao, S.H.; Cheng, M.M.; Zhao, K.; Zhang, X.Y.; Yang, M.H.; Torr, P. Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 652–662. [CrossRef] [PubMed]

43. Christiansen, P.H.; Kragh, M.F.; Brodskiy, Y.; Karstoft, H. UnsuperPoint: End-to-end Unsupervised Interest Point Detector and Descriptor. *arXiv* **2019**, arXiv:1907.04011.
44. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
45. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *CoRR* **2014**, abs/1412.6980.
46. Schmid, C.; Mohr, R.; Bauckhage, C. Evaluation of interest point detectors. *Int. J. Comput. Vis.* **2000**, *37*, 151–172. [CrossRef]
47. Lewis, J.P. Fast normalized cross-correlation. In Proceedings of the Vision Interface, Montreal, QC, Canada, 29 May–1 June 1995; Volume 10, pp. 120–123.
48. Sun, J.; Shen, Z.; Wang, Y.; Bao, H.; Zhou, X. LoFTR: Detector-free local feature matching with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 8922–8931.
49. Sarlin, P.E.; DeTone, D.; Malisiewicz, T.; Rabinovich, A. Superglue: Learning feature matching with graph neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 14–19 June 2020; pp. 4938–4947.
50. Balntas, V.; Lenc, K.; Vedaldi, A.; Mikolajczyk, K. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5173–5182.
51. Shi, J.; Tomasi, C. Good features to track. In Proceedings of the 1994 IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994; pp. 593–600.
52. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
53. Tan, M.; Le, Q. Efficientnetv2: Smaller models and faster training. In Proceedings of the International Conference on Machine Learning, Online, 18–24 July 2021; pp. 10096–10106.
54. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.; Tardós, J.D. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.