IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 May 2023, accepted 1 June 2023, date of publication 9 June 2023, date of current version 15 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3284463

RESEARCH ARTICLE

Enhancing Conventional Geometry-Based Visual Odometry Pipeline Through Integration of Deep Descriptors

MUHAMMAD BILAL, MUHAMMAD SHEHZAD HANIF, KHALID MUNAWAR, AND UBAID M. AL-SAGGAF, (Member, IEEE)
Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia
Corresponding author: Muhammad Shehzad Hanif (mshanif@kau.edu.sa)

The Deanship of Scientific Research (DSR) at King Abdulaziz University (KAU), Jeddah, Saudi Arabia has funded this Project under grant no (G:313-135-1443).

ABSTRACT Geometry-based Visual Odometry (VO) techniques are renowned in the fields of computer vision and robotics. They use methods from multi-view geometry to estimate camera motion from visual data obtained from one or more cameras. Tracking the camera motion precisely between different views is dependent on the correct estimation of correspondences between salient points of the views. In practice, geometry-based methods are found to be quite effective but do not perform well in challenging cases due to tracking failures caused by abrupt motion, occlusions, textureless and low-light scenes, etc. On the contrary, end-to-end learning from visual data using deep neural networks is an emerging area of research and deals with challenging cases successfully. Despite being computationally expensive, these methods do not outperform their counterparts in conditions favorable to geometry-based methods. Considering these facts in this work, our goal is to integrate deep descriptors to improve the correspondence between image points for tracking in a traditional geometry-based VO pipeline. We propose a simple stereo VO pipeline inspired by popular techniques found in the literature. Two conventional and four deep descriptors have been used in our experiments conducted on various image sequences of the challenging KITTI benchmark dataset. We have determined empirically that deep descriptors can effectively minimize drift in the VO estimates and produce better camera trajectories. The experimental results on the KITTI dataset demonstrate that our VO method performs at par with the state-of-the-art works reported in the literature.

INDEX TERMS Deep descriptors, deep neural networks, driverless vehicles, interest point detectors, mobile robots, visual odometry.

I. INTRODUCTION

Autonomous navigation is one desired functionality intended for driverless vehicles and mobile robots to move in an environment independently. To accomplish this challenging task, the vehicle (or robot) must be able to localize itself in the environment (localization problem) and build a map of its surroundings (mapping problem) using its sensors. Vision-based sensors are often the focus of interest as they provide rich information about the surroundings at a low cost. Visual Odometry (VO) deals with the estimation of vehicle motion using visual data from single or multiple cameras and helps in the localization of the vehicle in the environment [1]. In mobile robots, it is analogous to wheel odometry which provides the motion of a robot relative to its prior state or pose. However, the task is challenging as a VO algorithm is required to deal with abrupt motions, occlusions, variations in the scene illumination and low-textured environments [2], [3]. Visual Simultaneous Localization and Mapping (VSLAM), on the other hand, aims to build a map of the environment and localize the vehicle in the map at the same time using the visual data [4], [5]. Both VO and VSLAM

The associate editor coordinating the review of this manuscript and approving it for publication was Kathiravan Srinivasan.

share the same computational blocks for localization i.e., tracking of the camera across different views, but VSLAM has additional blocks like map management, loop closure and global trajectory correction. As for any odometry, an accurate VO algorithm is expected to reduce drift in the localization by limiting the tracking errors and therefore, helps in minimizing the need for loop closure and global trajectory correction [4].

Traditional VO methods [5], [6], [7], [8], [9], [10] employ geometry-based constraints and relationships to estimate the camera pose from one view (or image) to another. These methods provide reliable pose estimates but are subject to the scene conditions (e.g., moving objects, illumination and texture), and overlap between the two views. The exemplary performance of the deep learning-based methods in other challenging vision tasks like image recognition, object detection and tracking, scene segmentation, 3D reconstruction, etc., has also motivated the researchers to design and develop such methods for VO/VSLAM [2]. Therefore, in recent years, several deep learning-based methods [3], [11], [12], [13], [14] have been proposed for camera pose estimation with the focus to learn in an end-to-end manner from visual data. Their performance is better than the traditional VO methods in challenging low-textured and low-light scenes but is not good enough in comparison to their counterparts in favorable scene conditions [2], [13]. Moreover, these methods are also comparatively more computationally intensive.

Combining deep learning with geometry-based techniques for VO to gain the best out of both is an intriguing area of research and is the focus of this work. The two techniques can be fused in multiple ways but there are no established guidelines to achieve the best combination. Recent works like [15] and [13] propose to combine dense optical flow and depth networks with a geometry-based monocular VO technique which is quite expensive in terms of computations. In contrast, we have focused on the use of deep descriptors [16] with a traditional geometry-based VO pipeline for stereo camera setup. We have chosen the stereo setup instead of the monocular setup as the former is known to perform better at the estimation of metric scale required for pose estimation [1]. Moreover, descriptor matching is more efficient than the computation of dense optical flow using deep networks for tracking image points between different views. Our VO pipeline is influenced by popular VO/VSLAM strategies that have been put forth in the literature like C4VX [7], PTAM [10], stereo-PTAM [8], ORB-SLAM [9], and ORB-SLAM2 [5]. As mentioned earlier, tracking across views is the key component in VO and it is accomplished by establishing the correspondences between the views. In traditional methods like C4VX, PTAM, ORB-SLAM/ORB-SLAM2, these correspondences are determined by first computing hand-crafted descriptors like ORB around keypoints in one view and matching them with the help of a suitable distance function (e.g., Euclidean distance) in the other view. Recent works have shown that these hand-crafted descriptors perform poorly in challenging low-textured and low-light conditions while deep descriptors achieve better matching using the same distance function on benchmark datasets [17]. It is noteworthy that these deep descriptors are learned using image patches and the evaluation is performed on the benchmark datasets like UBC dataset [18] and Homography-patches (HPatches) dataset [19] for image patch verification and image matching. Although good enough for evaluating the descriptors in challenging scenes, yet, we believe these benchmark datasets do not account for scenarios faced by driverless vehicles and mobile robots. Challenges due to camera motion such as motion blur, occlusion, moving objects, etc. are not present at all in the UBC and HPatches benchmark datasets. Therefore, this work aims to evaluate four deep descriptors (TNet-TGLoss [20], PatchMatchNet [21], L2Net [22] and HardNet [23]) and compare them with two conventional descriptors (Scale Invariant Feature Transform (SIFT) [24] and ORB [25]) on the KITTI benchmark dataset [26] destined to evaluate algorithms for driverless vehicles. The four deep descriptors chosen in this work are diverse in terms of architecture, training methodology and loss function. For a fair comparison with conventional descriptors, we use the pre-trained models for deep descriptors instead of training them on the KITTI benchmark dataset. It is also important to note that the scenarios in the UBC and HPatches datasets are significantly different than that of the KITTI benchmark dataset. The main contributions and experimental findings of our work are summarized as follows:

- A geometry-based VO pipeline is proposed for stereo camera setup inspired by the building blocks of dominant VO/VSLAM approaches like C4VX, PTAM, and ORB-SLAM/ORB-SLAM2.
- We have employed deep descriptors in the proposed pipeline resulting in a hybrid approach that combines deep learning and the geometry-based VO approach.
- It is shown experimentally that the deep descriptors minimize drift in the VO estimates effectively and produce better camera trajectories compared to the conventional descriptors on the challenging KITTI benchmark dataset.
- In comparison with the five state-of-the-art methods (ORB-SLAM2 [5], convLSTM [27], Deep-VO-Feat [11], DF-VO [13], SF-VO [28]) on the KITTI benchmark dataset, our method achieves the lowest frame-to-frame translation error and yields competitive results in terms of frame-to-frame rotation error, drift in translation and rotation and absolute trajectory error.

The rest of the paper is structured as follows. Section II presents the relevant research works reported in the literature. The description of the VO pipeline and descriptors used in this work are provided in Section III and Section IV respectively. Experimental results along with the implementation notes are discussed in Section V. Finally, conclusions and future works are presented in Section VI.

II. RELATED WORKS

In this section, we describe the existing approaches and techniques related to VO which is a well-studied problem in the domains of computer vision and robotics. In addition to VO methods, VSLAM methods are also worth mentioning as both share the same pipeline. Traditionally, VO/VSLAM methods can be categorized as feature-based and direct using monocular and/or stereo cameras [1], [5], [10], [29], [30], [31]. These methods generally use image geometry for camera pose estimation. However, after the renaissance of neural networks, recent developments have focused on end-to-end deep learning-based methods [2], [3], [13] for camera pose estimation compared to traditional geometry-based methods. It is noteworthy that accurate scale estimation is a serious limitation in monocular setup and therefore, stereo setup usually yields better accuracy.

A. GEOMETRY-BASED METHODS

Single and multi-view geometry are well-known topics in computer vision. Feature-based methods [5], [6], [7], [8], [9], [10], [32], [33] for monocular, stereo and color-depth (RGB-D) camera setups have been proposed in the literature in the last two decades for pose estimation. These VO methods generally use a keypoint detector to determine the salient points (keypoints) in images, and feature vectors or descriptors are computed by considering the local region around each keypoint. Tracking of keypoints for establishing correspondences across different views (or image frames) is performed through descriptor matching. World points are also computed with the help of triangulation and maintained in the form of a map. Both map and tracked points are used to estimate the relative camera poses between views. Keypoint detector performance, type of descriptor and matching technique contribute to the accuracy of estimated poses between image frames.

Klein and Murray. in a seminal work, proposed parallel tracking and mapping (PTAM) [10] for estimation of the pose of a handheld camera in an unknown environment. In their work, tracking and mapping are executed as parallel tasks and optimized to estimate the camera pose in real-time. The ''features from the accelerated segment test'' (FAST) corner detector is used at multiple scales to detect keypoints in an image. Tracking of keypoints from one image frame to another is achieved through the image warping technique that warps patches around each keypoint in one view and matches them in the other by minimizing sum-of-absolute differences (SAD). World points are added to the map and computed for keyframes only, using the triangulation technique. Camera pose is estimated by minimization of reprojection error between the tracked keypoints and projection of the corresponding world points in the map. To reduce drift errors, Bundle adjustment (BA) [34] is performed locally on camera poses and world points in the map. In contrast to the previous work destined for monocular camera setup, the author in [32] presents a stereo VO algorithm for mobile robots. The stereo-rectified image pair is used to compute the disparity by matching pixels in the left and right images. Then, corners are detected in the rectified image pair using a standard corner detection algorithm, like the FAST corner detector, and pixel values in a fixed window around the corners are used as the descriptor. For tracking the corner points between consecutive stereo images, the descriptors are matched using the SAD score. To validate the matches, rigidity constraints on the corresponding world points are employed. The motion is estimated by minimizing the reprojection error between a set of left and right image keypoints and the projections of the corresponding world points. In a similar work, Geiger et al. employ corners as keypoints and image gradients in a local region around the keypoints as descriptors in their work on pose estimation and 3D reconstruction using stereo images [35]. The world points are computed using the triangulation using the calibration information of the stereo setup. For motion estimation, a reprojection error similar to [32] is employed and minimized with the help of a non-linear optimization scheme. A follow-up work [6] proposes a feature selection strategy based on the image bucketing method to reduce the computational complexity in descriptor matching for tracking purposes. However, contrary to the earlier work [35], Nistér five-point method [36] is used to estimate the essential matrix using the left image views only and is decomposed to get the rotation matrix. The translation vector is estimated by using the reprojection error in the stereo settings (using both left and right points and corresponding world point projections) to estimate the scale reliably.

Persson et al. propose C4VX in [7] which is a stereo VO method employing the FAST corner detector at multiple scales for keypoints and ''binary robust independent elementary features'' (BRIEF) descriptor. Tracking is performed by matching BRIEF descriptors of the left images of one view with those of the second view. Triangulation is performed to compute the world points using the stereo pair. The keypoints and corresponding world points are then used to estimate the camera pose with the help of a Perspective-n-point (PnP) solver [37]. Similar to PTAM, local BA is used to optimize the camera pose and world points. Mur-Artal et al. [9] proposed ORB-SLAM for monocular camera setup which is one of the widely-used VSLAM algorithms in the domain of robotics. Tracking, mapping and loop closure are three major tasks executed in parallel in the method. For all tasks, ''oriented FAST and rotated BRIEF'' (ORB) descriptors are used due to their excellent performance and reduced computational cost. The tracking is based on scale-aware ORB descriptor matching across image frames. World points are added to the map for each keyframe. To ensure long-term operation, world points and keyframes are reviewed regularly and are removed when they are not trackable. Camera pose estimation is based on the PnP solver using keypoints and corresponding world points. Local BA is employed to optimize the map and camera poses. Loop closure is performed with the help of a place recognition technique. A follow-up work [5] extends the ORB-SLAM to stereo and RGB-D camera setups.

Contrary to feature-based methods, direct methods formulate an energy function involving camera poses and camera intrinsics as parameters. The energy function is then minimized to compute the parameters. Moreover, the energy function relies on the dense representation of an image i.e., pixel values instead of keypoints and descriptors. Steinbrücker et al. [38] propose a direct method for the estimation of camera poses from RGB-D images. It is based on the minimization of a non-convex energy function aiming to find a warping function that warps one image view to another. The linearization of the energy function is performed under photoconsistency assumption for optimization. After convergence, the image warping function yields the relative camera pose. For large camera motions, a coarse-to-find approach at different scales of the image is applied. Similarly, in [30], the authors propose to minimize the photometric error where parameters like camera pose, camera intrinsics and world points are optimized jointly. Generally, the direct methods require a decent initialization for the parameters and are not resilient to illumination changes between frames [14].

B. DEEP LEARNING-BASED METHODS

Owing to the recent development in the field of deep neural networks, various end-to-end learning models for pose estimation have been suggested [2]. Recurrent convolutional neural networks are employed to estimate pose from color images in monocular settings in [12]. The inputs to the recurrent network are two consecutive frames and the network is trained to estimate the relative pose between them. The mean squared error between the ground truth pose and the output of the network is used as the loss function. The follow-up work of Jiao et al. [39] proposes to use a convolutional neural network (CNN) followed by a bi-directional long short-term memory (Bi-LSTM) for pose estimation. The CNN is used for feature extraction while the Bi-LSTM uses the sequential information between two frames for relative camera pose. The mean squared error defined in [12] is used as the loss function. Similarly, Pandey et al. [40] use a CNN for optical flow estimation which is fed to a Bi-LSTM network for pose estimation. The network is trained with the mean squared error between the ground truth pose and the predicted pose similar to [12]. Xue et al. [41] propose a framework based on a CNN and two additional modules namely memory and refining aiming to incorporate spatial and temporal information for camera tracking for monocular VO. A loss function based on both relative pose (local) error and trajectory (global) error is proposed for training the framework in an end-to-end fashion. A self-supervised monocular VO scheme (convLSTM) based on convolutional LSTM is proposed in [27] intending to adapt to unseen and dynamic test scenarios in an unsupervised manner. Similar to [41], the use of convolutional LSTM enables to focus on both spatial and temporal information for VO estimation. The self-supervised loss function used in their work combines appearance and depth information. In [42], Zhu et al. present a CNN for monocular VO composed of four streams each one concentrating on one quadrant of the optical flow aiming to exploit local visual cues. Each stream employs an attention mechanism to alleviate redundancy in the intermediate feature maps. The loss function is the mean square error between the ground truth pose and the predicted pose.

Contrary to the monocular approaches discussed above, Depth-VO-Feat [11] uses stereo pair for depth estimation and then uses it for pose estimation in an unsupervised way. For these tasks, a depth network and a pose network are trained jointly. The stereo images also help in determining the metric scale contrary to monocular images. Liu et al. [43] propose a stereo VO method using a deep neural network learning depth and pose jointly. Contrary to other approaches where a depth map is learned with the help of the ground truth depth map, they used ground truth pose to supervise the depth network in their joint learning framework.

In general, deep learning-based methods do not require camera calibration. However, the computational cost is significantly higher than their counterparts.

C. HYBRID METHODS

The geometry-based VO methods provide reliable estimates when correspondences are established accurately. Typically, sufficient texture and illumination are required in the scene to establish the correspondence correctly. In challenging environments, geometry-based methods do not perform well. On the other hand, deep VO methods provide promising results in challenging environments but perform inferior to their geometry-based counterparts, especially in scenarios favorable to geometry-based methods [2]. One interesting area of research is to combine recent advancements in the field of deep learning with geometry-based approaches for VO to maximize the benefits of both. However, there are numerous ways to achieve the fusion of the two approaches. For example, in [13], two CNNs, one for dense optical flow estimation and the other for depth estimation, are trained in a monocular setup. The dense optical flow and depth are fed to a conventional geometry-based VO approach. They have shown that their hybrid method achieved superior results. Similarly, DL-Hybrid [15] uses an optical flow network and a depth network with a traditional geometry-based method for a monocular VO system. Liu and Chen [28] proposed a sparse optical flow network for feature tracking and use a geometry-based method for pose computation. It is worth mentioning that they trained the optical flow network on synthetic data contrary to [13] and [15] and achieved competitive results. In a recent work on monocular VO [44], Cao et al. propose a joint learning framework composed of a depth network, an optical flow network, and a BA module. In their work, the traditional BA is modified to fit in the deep learning framework to improve the generalization ability of the VO method.

Our methodology presented in this work is motivated by these hybrid approaches. We study the influence of deep

descriptors in a geometry-based VO pipeline destined for a stereo setup, as deep descriptors have outperformed traditional descriptors in image matching and retrieval tasks on the benchmark datasets [17].

III. VISUAL ODOMETRY PIPELINE

In this section, we describe a feature-based VO pipeline for stereo images based on blocks from the conventional geometry-based pose estimation methods. Specifically. the pipeline is inspired by the dominant VO/VSLAM approaches proposed in the literature like C4VX [7], PTAM [10], stereo-PTAM [8], ORB-SLAM [9], and ORB-SLAM2 [5]. The complete algorithm is described in Algorithm 1. We assume the pinhole camera model in this work. The input to the VO pipeline is a sequence of n stereo-rectified image pairs $\{(I_j^{(l)}, I_j^{(r)})\}_{j=1}^{n}$ obtained through the calibration procedure where $I^{(l)}$ and $I^{(r)}$ denote left and right images respectively. In addition to the stereo-rectified images, the calibration procedure yields the stereo baseline (sb) and the camera intrinsic matrix ($\mathbf{K} \in \mathbb{R}^{3x3}$). The algorithm outputs camera poses $\mathbf{Q}_j, j = 1, \ldots, n$ where $\mathbf{Q}_j \in \text{SE}(3)$. In the following, we provide details of important blocks of the VO pipeline.

A. KEYPOINT DETECTOR AND DESCRIPTOR

A keypoint detector is applied to each stereo pair to output salient points denoted by the pair $(\mathbf{P}_j^{(l)}, \mathbf{P}_j^{(r)})$ where $\mathbf{P}_j^{(.)} = [\mathbf{p}_{1j}^{(.)}, \mathbf{p}_{2j}^{(.)}, \ldots, \mathbf{p}_{mj}^{(.)}]$ is composed of m keypoints. A d-dimensional descriptor is then used to represent a small patch around a keypoint efficiently with numeric values yielding a total of m descriptors denoted by $\mathbf{D}_j^{(l)}$ and $\mathbf{D}_j^{(r)}$ for left and right images respectively. The descriptor is assumed to be distinctive in nature and resilient to variations in scale, illumination, and viewpoint. In general, there is no criterion to choose the optimal detector and descriptor for a task and thus requires empirical evaluation. Therefore, in this work, we have considered different traditional and deep descriptor methods to evaluate their suitability for the VO pipeline in our experiments. The details of these detectors and descriptors are provided in section IV.

B. DESCRIPTOR MATCHING

An exhaustive approach to computing the pairwise distances between two sets of descriptors is employed in our VO pipeline which yields a distance matrix. The distance between a pair of descriptors is either squared Euclidean distance for real-valued descriptors or Hamming distance for binary descriptors. For matching i.e. establishing 2D-2D correspondences, the minimum distance in each column of the distance matrix must be less than a threshold which is set to 40% of the maximum distance. Moreover, we have considered unique matches only. In our pipeline, the *MatchDescriptor* function (see lines 12 and 17) returns the indices of the matched descriptors. The descriptor matching, on one hand, serves as a tracker that tracks the keypoints between $I_{j-1}^{(l)}$ and $I_j^{(l)}$ and on the other hand, is used to determine the correspondences between left and right stereo pair i.e., $(I_j^{(l)}, I_j^{(r)})$ for the construction of the world map.

Algorithm 1 Stereo Visual Odometry Algorithm

Input: Stereo-rectified image set: $\{(I_j^{(l)}, I_j^{(r)})\}_{j=1}^{n}$,
Camera intrinsic matrix: $\mathbf{K}$,
Stereo baseline: sb,
Window size: w
Result: Estimated camera poses: $\{\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_n\}$

1: Point set: $\mathcal{P} \leftarrow \varnothing$
2: Map: $\mathcal{M} \leftarrow \varnothing$
3: **for** $j = 1$ **to** n **do**
4: $\quad \mathbf{P}_j^{(l)} \leftarrow KeypointDetector\left(I_j^{(l)}\right)$
5: $\quad \mathbf{P}_j^{(r)} \leftarrow KeypointDetector\left(I_j^{(r)}\right)$
6: $\quad \mathbf{D}_j^{(l)} \leftarrow Descriptor\left(I_j^{(l)}, \mathbf{P}_j^{(l)}\right)$
7: $\quad \mathbf{D}_j^{(r)} \leftarrow Descriptor\left(I_j^{(r)}, \mathbf{P}_j^{(r)}\right)$
8: $\quad$ **if** $j = 1$ **then**
9: $\quad\quad \mathbf{Q}_j \leftarrow \mathbf{I}_{4x4}$
10: $\quad\quad \mathcal{P} \leftarrow \mathbf{P}_j^{(l)}$
11: $\quad$ **else**
12: $\quad\quad L \leftarrow MatchDescriptor\left(\mathbf{D}_{j-1}^{(l)}, \mathbf{D}_j^{(l)}\right)$
13: $\quad\quad \mathbf{Q}_j \leftarrow SolvePnP\left(\mathbf{P}_j^{(l)}, \mathcal{M}, L, \mathbf{K}\right)$
14: $\quad\quad \mathbf{Q}_j \leftarrow PoseRefinement\left(\mathbf{Q}_j, \mathbf{P}_j^{(l)}, \mathcal{M}, L, \mathbf{K}\right)$
15: $\quad\quad \mathcal{P} \leftarrow UpdatePointSet\left(\mathbf{P}_j^{(l)}, L\right)$
16: $\quad$ **end**
17: $\quad S \leftarrow MatchDescriptor\left(\mathbf{D}_j^{(l)}, \mathbf{D}_j^{(r)}\right)$
18: $\quad \mathcal{M} \leftarrow UpdateMap\left(I_j^{(l)}, I_j^{(r)}, S, \mathbf{Q}_j, \mathbf{K}, sb\right)$
19: $\quad$ **if** $\text{mod}(j, w) = 0$ **then**
20: $\quad\quad (\mathcal{M}, \mathbf{Q}_{\mathcal{V}(j)}) \leftarrow LocalPoseMapOptimization\left(\mathcal{P}, \mathcal{M}, \mathbf{Q}_{\mathcal{V}(j)}, \mathbf{K}\right)$
21: $\quad$ **end**
22: **end**

C. POINT SET AND WORLD MAP

The keypoints $(\mathbf{P}_j^{(l)})$ and tracking information (L) are stored in the form of a point set denoted by $\mathcal{P}$ and are updated at each image frame (see lines 10 and 15). Additionally, the local 3D scene structure information i.e. world points with respect to the current pose $(\mathbf{Q}_j)$ and corresponding tracking information (S) are stored in a world map (denoted by $\mathcal{M}$) and are also updated using the untracked points at each image frame (see line 18). To compute the world points, we first compute the disparity map using the Semi-global matching algorithm [45] and then employ the reprojection matrix [46]. We recall that the reprojection matrix uses the principal point and the focal length from the camera intrinsic matrix ($\mathbf{K}$) and the stereo baseline (sb). It is noteworthy that the new world

points are added without further verification following the strategy proposed in [9] and [5].

D. POSE ESTIMATION

The 3D-2D correspondences can now be obtained from the tracking information stored in the point set ($\mathcal{P}$) and the map ($\mathcal{M}$). Pose estimation from 3D-2D correspondences is an optimization problem where the aim is to determine the camera pose by minimizing the reprojection error between the 3D world point and the 2D image point. Let $(\mathbf{x}_k, \mathbf{p}_k^{(l)})$ represent the kth 3D-2D correspondence at the jth frame (or view) where $\mathbf{x}_{kj} \in \mathcal{M}$ and $\mathbf{p}_{kj}^{(l)} \in \mathcal{P}$. Moreover, let $\tilde{\mathbf{p}}_{kj}^{(l)}$ denotes the projection of the 3D world point $\mathbf{x}_k$, we can write the optimization problem as:

$$\underset{\mathbf{Q}_j}{\operatorname{argmin}} \sum_k \|\mathbf{p}_{kj}^{(l)} - \tilde{\mathbf{p}}_{kj}^{(l)})\|_2^2. \tag{1}$$

where $\|.\|_2$ is the l_2 norm.

The above optimization problem is also known as Perspective-n-Point (PnP) problem. It can be solved by using the well-known Perspective-n-Point (PnP) solvers like [37] and [47] by employing the image geometry. Specifically, We use P3P solver [47] which requires a minimum of three correspondences, with random sample consensus (RANSAC) for estimating the camera pose. The purpose of RANSAC is to deal with the outliers i.e., noisy correspondences as the P3P solution is not optimal in the presence of outliers.

For the first image frame, the camera pose is assumed to be aligned with the world frame. The camera pose, in subsequent frames, is determined by the P3P solver. Once the initial estimate of the camera pose is determined in an image frame, it is refined with the help of the motion-only Bundle Adjustment (BA) technique by considering all inliers i.e. the largest consensus set (see line 14) as proposed in [7] and [5]. In general, the BA problem is to minimize the reprojection error in (1) which is minimized efficiently with the help of the Levenberg-Marquardt (LM) algorithm [34]. It is important to note that, in the case of motion-only BA, the camera pose $\mathbf{Q}_j$ is optimized and points are fixed. The maximum number of iterations is fixed to 20 in our implementation for pose refinement.

E. LOCAL POSE-MAP OPTIMIZATION

As described previously, the camera pose at jth image frame ($\mathbf{Q}_j$) is determined with the help of 3D-2D correspondences and the PnP solver. However, the 3D world points are also dependent on $\mathbf{Q}_{j-1}$. Thus, the current camera pose is directly dependent on the previous camera poses. Estimation errors in camera pose propagate and lead to drifting over time. To alleviate the drifting problem, BA is introduced at regular intervals in the VO pipeline where both camera poses and map points in the local neighborhood ($\mathcal{V}$) of the jth image frame are optimized jointly (see line 20). Let $\Omega(\mathbf{Q}_j, \mathbf{x}_k)$ denote a reprojection function that projects the 3D world point $\mathbf{x}_k$ to jth camera view. Moreover, let $\mathbf{p}_{kj}^{(l)}$ represent the kth 2D image point in jth camera view. We can write the local BA optimization problem in the neighborhood of jth camera view defined by a set $\mathcal{V}$ of views as the minimization of the following reprojection error:

$$\min \sum_k \sum_{i \in \mathcal{V}(j)} u_{ki} \|\mathbf{p}_{ki}^{(l)} - \Omega(\mathbf{Q}_i, \mathbf{x}_k)\|_2^2. \tag{2}$$

where $u_{ki} = 1$ if kth world point is visible in ith camera view and 0 otherwise.

As BA is computationally intensive, only 10 iterations of the LM algorithm are performed. Furthermore, to improve the quality of 3D-2D correspondences, the world points with sufficiently large reprojection errors are removed from the map. This step ensures that the good quality world points are retained.

IV. DESCRIPTORS

In this section, we describe in detail the different conventional and deep descriptors used in this work. Specifically, we consider two widely used conventional descriptors known as SIFT [24] and ORB [25]. Both SIFT and ORB are pseudo-standards in the domain of computer vision due to their excellent performance in several applications including stereo matching, object recognition, object tracking, 3D reconstructions, place recognition, VO, and VSLAM, etc. [5], [48], [49], [50]. In recent years, descriptor learning from data is the focus of researchers working in the field of computer vision owing to rapid advancement in deep learning methods. We have considered four different deep descriptors in this work namely TNet-TGLoss [20], PatchMatchNet [21], L2Net [22] and HardNet [23]. They have achieved significant results while outperforming the conventional descriptors on image patch verification and image retrieval tasks [17]. Table 1 summarizes the descriptors used in this work, their dimensionality and the number of parameters in the network for deep descriptors. Details of both conventional and deep descriptors are presented in the following subsections. It is noteworthy that the above-mentioned deep descriptor learning methods do not incorporate an interest point detector contrary to the conventional descriptors. However, without loss of generality, any interest point detector can be employed. It is also worth mentioning that the research works [20], [21], [22] have described several networks for deep descriptor learning including two-channel and center-surround networks. In this work, we have focused on the single-branch variants of the proposed networks because they need fewer computations compared to two-channel and center-surround networks. The implementation details are presented in section V-C.

A. SCALE INVARIANT FEATURE TRANSFORM (SIFT)

Scale Invariant Feature Transform (SIFT) [24], proposed by David Lowe, is one of the widely used descriptors employed in several computer vision applications including image retrieval, stereo matching, object recognition, 3D reconstruction, place recognition, object tracking, etc. [48], [49], [51], [52], [53], [54], [55]. The success of the SIFT descriptor

TABLE 1. Conventional and deep descriptors used in the VO pipeline.

Descriptor	Dimension	Number of Parameters
SIFT [24]	128D	-
ORB [25]	32D	-
TNet-TGLoss [20]	256D	≈ 1M
PatchMatchNet [21]	512D	≈ 0.6M
L2Net [22]	128D	≈ 1.3M
HardNet [23]	128D	≈ 1.3M

is mainly due to its resilience to illumination, scale, and viewpoint variations. The computation of the SIFT descriptor is composed of two stages known as keypoint detection and computation of Histogram of Oriented Gradients (HOG). The keypoint locations are the extrema in the scale-space pyramid that is constructed using the Difference of Gaussian (DOG). A fixed region of size 16 × 16 around each keypoint location is considered to describe the keypoint. An 8-bin HOG is computed for each 4 × 4 non-overlapping subregion in the fixed region to describe the keypoint. All 16 histograms are concatenated to form a final 128D real-valued vector. The descriptor is normalized to achieve an illumination invariant descriptor.

B. ORIENTED FAST AND ROTATED BRIEF (ORB)

Oriented FAST and Rotated BRIEF (ORB) descriptor was proposed by Rublee et al. [25] as an alternative to SIFT and speeded up robust feature (SURF) descriptors with an emphasis on reducing the computational complexity. The descriptor is binary in nature and is based on the computationally efficient FAST corner detector [56] and BRIEF descriptor [57]. Generally, multiscale FAST corners are detected on the image pyramid determined by user-defined scales and serve as keypoints. The orientations of keypoints are determined with the help of the intensity centroid method. The rotation invariance is improved by computing moments in a circular region of fixed size around the keypoint. Then, for each keypoint, the rotated BRIEF descriptor is computed by taking into account the orientation of the keypoint. Being invariant to rotation, scale, viewpoint, and illumination, the ORB descriptor has been employed successfully in fields like object recognition, image matching, structure from motion, and VSLAM, etc. [5], [9], [50]. In [25], the authors propose to use a circular region of 31 × 31 pixels around the keypoint and a total of 8 scales with a scale factor of 1.2. These settings result in a 32D integer-valued (i.e. 256 bit) descriptor.

C. TRIPLET NETWORK WITH GLOBAL LOSS (TNet-TGLoss)

In [20], Kumar et al. proposed a descriptor learning framework trained with a custom loss function that combined a novel loss function called global loss and the triplet loss [58]. The learning framework consists of convolution and max-pooling layers with rectified linear units (ReLU) as activation functions after convolution layers. The complete network architecture called TNet-TGLoss is shown in Fig. 1. The input to the network is a grayscale patch of 64 × 64 pixels and the output is a 256D descriptor after the application of l_2 normalization. The arguments to convolution (conv) layers

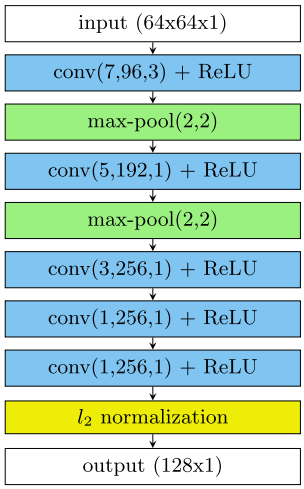


FIGURE 1. TNet-TGLoss network architecture.

are filter size, number of filters, and stride respectively. Moreover, the input arguments to pooling (max-pool/avg-pool) layers are its spatial size and stride respectively. The number of training parameters in the network is 1M approximately. They trained the network from scratch using the 250,000 image triplets from the UBC dataset [18] with the custom loss function. Data augmentation (rotation, flipping the image horizontally and vertically) is also employed to achieve robust descriptors. Though the network can be trained using the triplet loss function, it requires a large number of triplets sampled from the training set which may not be feasible due to the enormous number of possible triplets. Moreover, the network may learn to map all inputs to a single point in the output space. To alleviate these problems, Kumar et al. proposed combining the triplet loss with the global loss. They showed that the distances between descriptors of similar and dissimilar pairs follow two distinct distributions and used the respective means and variances to form the global loss function. Specifically, the loss aims to: (1) minimize the mean of distances of similar pairs; (2) maximize the mean of distances of dissimilar pairs; (3) minimize the variances of the two distributions. The trained network is shown to perform excellently on the image patch verification task on the UBC benchmark dataset.

D. PATCH MATCH NETWORK (PatchMatchNet)

In one of our previous works related to descriptor learning [21], Patch Match Network (PatchMatchNet) was proposed, It is based on densely connected convolution layers contrary to plain convolution layers in a CNN for image patch matching task. It has been shown that the network based on densely connected convolution layers learns a better representation of the inputs compared to its counterpart with plain convolution layers. The network is composed of convolution (conv), concatenation (concatenate), max-pooling (max-pool), and average pooling (avg-pool) layers (see Fig. 2). Each convolution layer is followed by a batch normalization

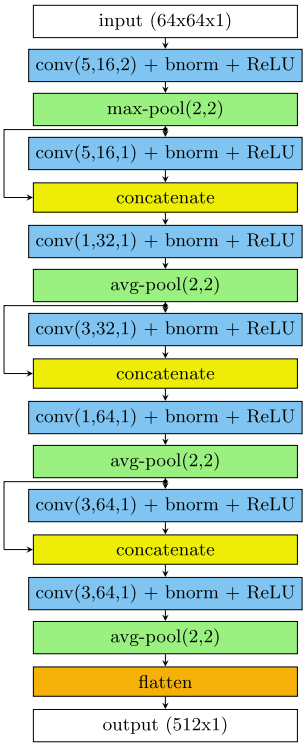


FIGURE 2. PatchMatchNet network architecture.

(bnorm) layer and ReLU activation function. The dense convolution layer is achieved by a convolution layer and a concatenation operation applied to the input and output of the convolution layer. The total number of training parameters in the network is 0.6M approximately. For descriptor learning, a Siamese network is trained with the help of image patch pairs. Then, the outputs of the two branches of the network are concatenated and fed to a 2-layer fully connected network (called top network) to learn a similarity score for a pair of inputs. The hinge loss is used for training the Siamese and top networks simultaneously. Once trained, the output of one branch of the Siamese network is a 512D descriptor representing the input grayscale patch of 64 × 64 pixels. The network is trained from scratch on 500,000 image patch pairs from the UBC dataset [18]. Data augmentation (random flip and random rotation) is also employed during training. The l_2 normalization is not part of the network rather it is done separately after the computation of the descriptor. The network is shown to outperform networks with plain convolution layers on the image patch verification task of the UBC dataset. Moreover, the generalization capability of the descriptor is shown on the HPatches benchmark dataset [19] which includes tasks like patch verification and image retrieval.

E. L2Net

L2Net [22] is a stack of plain convolution layers with filter sizes of 3 × 3. The number of filters is varied from 32 to

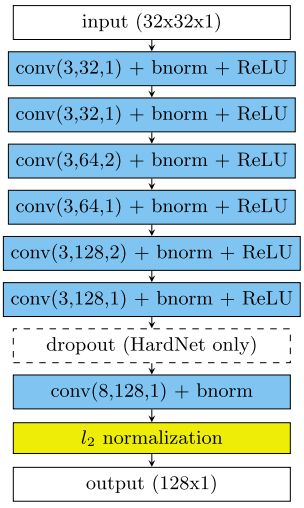


FIGURE 3. L2Net and HardNet both share the same architecture. The HardNet uses a dropout layer before the final convolution layer.

128 in the seven convolution layers of the network as can be seen in Fig. 3. The batch normalization (bnorm) layer and ReLU activation function are used after each convolution layer except the final layer. The final layer employs l_2 normalization instead of ReLU to output a 128D real-valued descriptor. The input to the network is a grayscale patch of 32 × 32 pixels. The network has 1.3M training parameters approximately. For descriptor learning, the authors propose a progressive sampling strategy for training that enables the network to see a large number of image patch pairs in a few epochs. In the proposed sampling strategy, in one mini-batch of size b, a certain image patch has only one more image with the same label. Thus, each batch has $b/2$ unique labels yielding b similar pairs and $b^2 - b$ dissimilar pairs. Therefore, the network can be trained on a large number of pairs within a few epochs. The loss function is composed of three error terms. The first term aims to minimize the relative Euclidean distances between descriptors of similar pairs and maximize the relative Euclidean distances between descriptors of dissimilar pairs. The second term aims to achieve compactness of the descriptor by minimizing the correlation between different dimensions of the descriptor. The third term aims to bring the intermediate feature maps of similar pairs together while pulling apart those of dissimilar pairs. During training, data are augmented with the help of random rotations and random flips. The L2Net has achieved remarkable performance on image patch verification and image retrieval tasks on the benchmark datasets including UBC and HPatches datasets.

F. HardNet

Mishchuk et al. [23] proposed HardNet for descriptor learning from image patches by adopting the L2Net architecture with a different training methodology. They proposed to use a dropout layer before the final convolution layer of L2Net architecture for the regularization purpose (see Fig. 3). Similar to L2Net, the input to the network is a grayscale patch of

32×32 pixels and the output is 128D real-valued descriptor. Moreover, the mini-batch is constructed in the same way as described in the case of L2Net. The loss function is based on the Euclidean distance between image patch pairs in a mini-batch and contrary to L2Net, does not consider the compactness and intermediate features maps. For training, a triplet loss is used which aims to minimize the distance between similar pairs for each distinct $b/2$ image patch in the mini-batch while maximizing the distance to the closest dissimilar pair involving the same distinct image patch. The technique called "hardest-within-batch" mining scheme has performed remarkably on the UBC and HPatches benchmark datasets for image patch verification and image retrieval tasks. Data augmentation similar to the above-mentioned networks is also considered during the training.

V. EXPERIMENTAL RESULTS

In this section, we present the benchmark dataset followed by the evaluation metrics and implementation details. Both quantitative and qualitative results for our VO pipeline using conventional and deep descriptors are provided. Moreover, a comparison with the state-of-the-art methods is also presented.

A. KITTI VISUAL ODOMETRY DATASET

The KITTI Visual Odometry dataset [26] introduced in 2012 is a benchmark dataset for evaluating VO and VSLAM algorithms. It is one of the most used datasets because it is a large-scale dataset representing a realistic outdoor driving scenario. The dataset provides a total of 22 stereo sequences (labeled as 00 to 21) which are collected by driving a station wagon in an urban environment for a total distance of 39.2 km. The vehicle is equipped with a stereo camera setup, a 3D LIDAR and a GPS/IMU localization unit. The stereo setup is composed of 2 grayscale and 2 color cameras and images are recorded at a rate of 10 frames per second. The camera setup has an image resolution of 1392×512 pixels and the stereo baseline is 54 cm. The dataset provides stereo-rectified images and corresponding calibration data. Additionally, the ground truth of 11 sequences (from sequence 00 to sequence 10) representing vehicle trajectory is provided. In this work, we have used stereo-rectified grayscale images from the 11 sequences with corresponding ground truth.

B. EVALUATION METRICS

In this work, we have employed the KITTI odometry evaluation criteria [26] to evaluate quantitatively the influence of descriptors on the VO algorithm. The evaluation criteria are based on the translation and rotation errors computed using the ground truth and estimated trajectories. Generally, two types of errors, relative and absolute errors in the trajectory, are computed and discussed below. It is noteworthy that the relative error determines the drift in the pose locally as the vehicle moves in an environment while absolute error denotes the localization accuracy in the global sense.

Let $\mathbf{G}_1, \ldots, \mathbf{G}_n \in \mathrm{SE}(3)$ denote the ground truth poses for n frames in a sequence. The relative pose error between the ground truth and estimated camera poses at a time instant j over a fixed interval δ is defined as:

$$\mathbf{E}_j = \left[\begin{array}{c|c} \mathbf{R}_{\mathbf{E}_j} & \mathbf{t}_{\mathbf{E}_j} \\ \hline \mathbf{0}^T & 1 \end{array}\right] = (\mathbf{G}_j^{-1}\mathbf{G}_{j+\delta})^{-1}(\mathbf{Q}_j^{-1}\mathbf{Q}_{j+\delta}). \tag{3}$$

where T denotes the transpose of a vector.

Using the above equation, rotation error (E_{rot}) and translation error (E_{trans}) at time instant j are computed as follows:

$$E_{rot_j} = \arccos\left(\frac{trace(\mathbf{R}_{\mathbf{E}_j}) - 1}{2}\right). \tag{4}$$

$$E_{trans_j} = \|\mathbf{t}_{\mathbf{E}_j}\|_2. \tag{5}$$

In the KITTI odometry evaluation criteria, (3) to (5) are employed to compute two sets of relative rotation and translation errors. In the first set, all possible sub-sequences of length in [100 m, 200 m, ..., 800 m] are considered by adjusting the indices i and δ in (3) to compute the sub-sequence rotation and translation errors. The average translation and rotation errors denoted as $t_{err}(\%)$ and $r_{err}(\%)$ [deg/100m] respectively are then computed for a sequence by averaging over all possible sub-sequences. In the second set, frame-to-frame pose error known as relative pose error (RPE) is computed by setting δ as 1 in (3). The average translation and rotation errors denoted as t_{RPE} [m] and r_{RPE} [deg] are then computed for a sequence using (4) and (5). In addition to the relative errors, absolute trajectory error (ATE) [m] is computed by considering the translation vectors of ground truth poses $\mathbf{t}_{\mathbf{G}_j}$ and estimated pose $\mathbf{t}_{\mathbf{Q}_j}$. Mathematically, the ATE at time instant j is computed as follows:

$$\mathrm{ATE}_j = \|\mathbf{t}_{\mathbf{G}_j} - \mathbf{t}_{\mathbf{Q}_j}\|_2. \tag{6}$$

The average value of ATE is computed for an entire sequence to report the localization accuracy.

C. IMPLEMENTATION DETAILS

The VO pipeline, SIFT, and ORB are implemented using MATLAB®computer vision and image processing toolboxes. The TNet-TGLoss and L2Net rely on the MatConvNet toolbox for MATLAB® [59]. The PatchMatchNet and HardNet use TensorFlow [60] and PyTorch [61] respectively. For deep descriptors, repositories mentioned by the respective authors are used to get the pre-trained models. For a fair comparison, models trained on the *Liberty* subset of the UBC benchmark dataset [18] are used in our experiments. We have used the multiscale FAST detector which is native to the ORB as the keypoint detector for all deep descriptors. A total of 1000 keypoints spread evenly on the input image are considered for descriptor computations for both conventional and deep methods. In the VO pipeline, the local BA is performed after every w frames (see algorithm 1) which is set to 5 in our experiments. A desktop PC equipped with i7-4770 CPU, 12GB RAM, and 11GB GTX 1080Ti graphic card running

TABLE 2. Quantitative VO results using conventional and deep descriptors on sequences 00 to 10 of the KITTI dataset.

Descriptor	Metric	00	01	02	03	04	05	06	07	08	09	10	Average
ORB	t_{err}	3.770	14.470	3.090	2.890	2.160	2.650	2.630	2.290	2.780	2.970	2.300	3.818
	r_{err}	1.620	3.890	1.230	0.990	1.030	1.260	1.170	1.640	1.270	0.980	1.640	1.520
	ATE	90.940	369.960	173.490	14.480	6.470	39.980	**9.680**	11.740	53.170	35.890	24.610	75.492
	t_{RPE}	0.033	0.452	0.041	0.035	0.046	0.026	0.038	0.023	0.036	0.043	0.032	0.073
	r_{RPE}	0.092	0.452	0.090	0.071	0.055	0.065	0.065	0.062	0.074	0.082	0.083	0.108
SIFT	t_{err}	5.350	**8.860**	6.360	4.870	3.460	4.260	4.140	4.650	4.900	5.230	5.360	5.222
	r_{err}	2.350	2.060	2.420	1.920	2.500	2.070	1.840	3.110	2.260	1.820	2.520	2.261
	ATE	135.860	381.150	309.990	27.950	13.060	66.270	24.250	21.910	145.620	75.170	60.790	114.729
	t_{RPE}	**0.024**	**0.174**	**0.029**	**0.027**	0.039	**0.020**	**0.029**	**0.017**	**0.029**	**0.031**	**0.021**	**0.040**
	r_{RPE}	**0.073**	0.156	**0.071**	**0.055**	0.057	**0.049**	**0.054**	**0.049**	**0.055**	**0.062**	**0.060**	**0.067**
TNet-TGLoss	t_{err}	**3.060**	25.600	2.940	3.300	2.000	2.720	2.820	3.230	2.850	2.780	2.160	4.860
	r_{err}	**1.270**	1.730	1.140	1.360	1.020	1.300	1.240	2.020	1.140	1.100	1.110	1.312
	ATE	**71.100**	408.900	**117.130**	15.100	6.860	37.380	12.810	15.120	**31.070**	36.560	21.740	70.343
	t_{RPE}	0.031	0.577	0.036	0.032	0.042	0.026	0.036	0.023	0.035	0.039	0.026	0.082
	r_{RPE}	0.089	0.205	0.088	0.070	0.057	0.067	0.064	0.061	0.071	0.082	0.080	0.085
PatchMatchNet	t_{err}	3.160	32.600	**2.750**	2.780	1.650	**1.900**	2.930	**2.100**	**2.440**	2.780	**1.710**	5.164
	r_{err}	1.400	1.600	1.140	1.190	0.840	**1.180**	1.230	**1.450**	**1.080**	1.110	1.140	1.125
	ATE	78.230	530.880	120.060	11.520	5.400	**27.370**	13.940	**9.420**	38.610	33.980	**14.780**	80.381
	t_{RPE}	0.032	0.720	0.037	0.033	0.042	0.027	0.037	0.024	0.036	0.038	0.028	0.096
	r_{RPE}	0.092	0.202	0.090	0.072	0.056	0.070	0.064	0.062	0.075	0.081	0.082	0.086
L2Net	t_{err}	3.240	17.920	3.000	2.810	1.850	2.510	2.520	2.720	2.730	**1.780**	2.170	3.932
	r_{err}	1.420	**1.380**	**1.100**	1.240	0.930	1.240	1.030	1.810	1.150	**0.670**	1.100	1.188
	ATE	84.620	272.630	136.630	12.410	5.180	38.290	10.290	12.320	56.160	**21.160**	22.720	**61.128**
	t_{RPE}	0.029	0.412	0.033	0.028	0.038	0.024	0.032	0.021	0.033	0.034	0.024	0.064
	r_{RPE}	0.085	0.150	0.082	0.065	0.051	0.062	0.058	0.058	0.068	0.070	0.074	0.075
HardNet	t_{err}	3.490	14.300	3.010	**2.210**	**1.580**	2.360	**2.510**	2.590	2.920	2.810	1.940	**3.611**
	r_{err}	1.480	1.500	1.190	**0.710**	**0.790**	1.310	**1.000**	1.810	1.250	0.980	**1.030**	**1.186**
	ATE	83.580	**245.560**	154.730	**9.270**	**4.940**	38.980	11.100	11.250	57.470	36.130	20.560	61.234
	t_{RPE}	0.028	0.334	0.033	0.030	**0.037**	0.024	0.031	0.021	0.033	0.033	0.024	0.057
	r_{RPE}	0.084	**0.145**	0.084	0.070	**0.051**	0.062	0.057	0.058	0.068	0.072	0.073	0.075

Note: The best and runner-up are shown in bold and underlined text respectively.

the Ubuntu operating system serves as the computational platform in all our experiments.

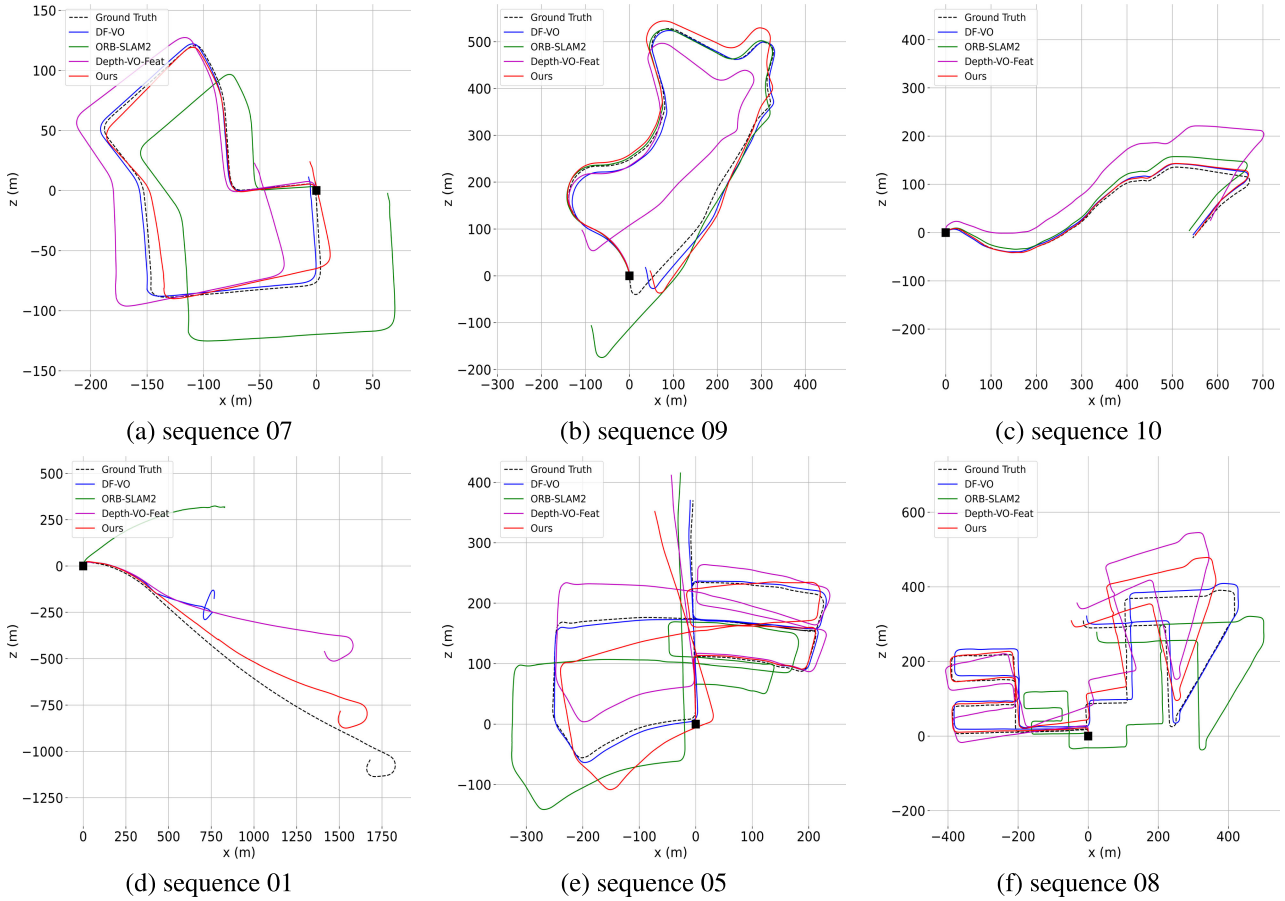
D. VISUAL ODOMETRY RESULTS

We evaluate the performance of six descriptors (ORB, SIFT, TNet-TGLoss, PatchMatchNet, L2Net, HardNet) using the geometry-based VO pipeline of section III and the above-mentioned evaluation metrics. The results in terms of relative errors i.e., $t_{err}(\%)$, $r_{err}(\%)$ [deg/100m], t_{RPE} [m] and r_{RPE} [deg] and absolute errors i.e., ATE [m] on 11 sequences of the KITTI dataset are presented in Table 2. The winner and runner-up for each sequence and each metric are also shown in bold and underlined text respectively. On comparing conventional descriptors only, ORB achieves lower average translation drift (t_{err} = 3.818%), average rotation drift (r_{err} = 1.520%) and average ATE (75.492 [m]) than SIFT. But, SIFT performs better than ORB in terms of average frame-to-frame relative errors i.e. t_{RPE} and r_{RPE} achieving relative pose errors of 0.040 [m] and 0.067 [deg] respectively. Thus, it can be concluded that ORB is better than SIFT at dealing with the drift problem in the VO estimation over longer periods. On comparing deep descriptors only, HardNet achieves lower average translation drift (t_{err} = 3.611%) and average rotation drift (r_{err} = 1.186%) than TNet-TGLoss, PatchMatchNet and L2Net. In terms of ATE, L2Net is slightly better than HardNet achieving an average ATE of 61.128 [m] but is significantly better than PatchMatchNet (80.381 [m]) and TNet-TGLoss (70.343 [m]). Considering average frame-to-frame relative errors, both HardNet (0.057 [m] and 0.075 [deg]) and L2Net (0.064 [m] and 0.075 [deg]) achieve similar results and are better than PatchMatchNet and TNet-TGLoss. Moreover, it is noted that PatchMatchNet, L2Net and HardNet attained best or runner-up positions on 9 out of 11 sequences while TNet-TGLoss reached the top two positions in 5 sequences only in one or more evaluation metrics. Thus, it can be stated that a modified CNN architecture employed by the PatchMatchNet and/or a better sampling strategy during training employed by the L2Net and HardNet yielded better deep descriptors. It is also noteworthy that PatchMatchNet has only 0.6M parameters compared to 1.3M of L2Net and HardNet and achieves competitive results but yields high dimensional descriptors (see Table. 1). This makes L2Net and HardNet promising choices due to their superior performances while yielding only 128D descriptors. Considering the average errors for all descriptors, the HardNet achieves the best t_{err} (3.611%) and r_{err} (1.186%). Compared to HardNet, the runner-up ORB descriptor achieves a translation error of 3.818%, lagging by 0.21%, while the L2Net achieves a slightly higher rotation error of 1.188%, lagging by just 0.002%. The L2Net attains an average ATE of 61.128 [m] computed over all sequences and outperforms all others. It is followed by the HardNet achieving an ATE of 61.234 [m]. In terms of relative error (t_{RPE} and r_{RPE}), the SIFT outperforms all on 10 out of 11 sequences and achieves the lowest relative pose errors on average. It is followed by the HardNet achieving very close relative errors and lags

TABLE 3. Quantitative comparison with existing methods on sequences 00 to 10 of the KITTI dataset.

Method	Metric	00	01	02	03	04	05	06	07	08	09	10	Average
ORB-SLAM2 [5] (w/o loop closure)	t_{err}	11.430	107.570	10.340	**0.970**	*1.300*	9.040	14.560	9.770	11.460	9.300	2.570	17.119
	r_{err}	0.580	**0.890**	**0.260**	**0.190**	0.270	**0.260**	**0.260**	0.360	**0.280**	**0.260**	**0.320**	**0.357**
	ATE	*40.650*	502.200	*47.820*	**0.940**	*1.300*	29.950	40.820	16.040	43.090	38.770	*5.420*	69.727
	t_{RPE}	0.169	2.970	0.172	*0.031*	*0.078*	0.140	0.237	0.105	0.192	*0.128*	0.045	0.388
	r_{RPE}	0.079	**0.098**	0.072	0.055	*0.079*	0.058	0.055	0.047	0.061	0.061	0.065	**0.066**
convLSTM [27]	t_{err}	14.210	21.360	16.210	18.410	9.080	24.820	9.770	12.850	27.100	15.210	25.630	17.695
	r_{err}	5.930	4.620	2.600	0.890	4.410	6.330	3.580	2.300	7.810	5.280	7.690	4.676
Depth-VO-Feat [11]	t_{err}	6.230	23.780	6.590	15.760	3.140	4.940	5.800	6.490	5.450	11.890	12.820	9.354
	r_{err}	2.440	*1.750*	2.260	10.620	2.020	2.340	2.060	3.560	2.390	3.600	3.410	3.314
	ATE	64.450	203.440	85.130	21.340	3.120	*22.150*	14.310	15.350	*29.530*	52.120	24.700	48.695
	t_{RPE}	*0.084*	0.547	*0.087*	0.168	0.095	**0.077**	*0.079*	*0.081*	*0.084*	0.164	0.159	*0.148*
	r_{RPE}	0.202	0.133	0.177	0.308	0.120	0.156	0.131	0.176	0.180	0.233	0.246	0.187
DF-VO [13]	t_{err}	1.960	56.760	2.380	2.490	1.030	**1.100**	**1.030**	**0.970**	1.600	2.610	*2.290*	*6.747*
	r_{err}	*0.600*	13.930	*0.550*	0.390	**0.250**	0.300	0.300	**0.270**	0.320	0.290	0.370	1.597
	ATE	11.340	484.860	21.160	2.040	0.860	**3.630**	**2.530**	**1.720**	**5.660**	10.880	3.720	*49.855*
	t_{RPE}	**0.027**	*1.203*	0.033	**0.023**	**0.036**	0.020	**0.024**	**0.018**	**0.032**	0.056	*0.047*	0.138
	r_{RPE}	**0.055**	0.773	**0.046**	**0.037**	**0.030**	**0.035**	**0.029**	**0.030**	**0.037**	**0.037**	**0.043**	*0.105*
SF-VO [28]	t_{err}	**1.180**	*21.600*	**1.280**	2.070	**0.540**	1.110	1.270	2.050	**1.590**	**1.100**	**1.240**	**3.185**
	r_{err}	**0.460**	10.400	0.430	*0.650*	*0.670*	*0.480*	*0.530*	*1.280*	*0.520*	*0.400*	*0.500*	*1.484*
	ATE	**7.010**	**196.960**	**11.160**	*3.530*	**0.490**	4.350	2.590	4.310	6.690	**4.070**	**1.880**	**22.095**
Ours (HardNet Descriptor)	t_{err}	*3.490*	**14.300**	*3.010*	*2.210*	1.580	*2.360*	*2.510*	*2.590*	*2.920*	*2.810*	1.940	3.611
	r_{err}	1.480	1.500	1.190	0.710	0.790	1.310	1.000	1.810	1.250	0.980	1.030	1.186
	ATE	83.580	*245.560*	154.730	9.270	4.940	38.980	*11.100*	*11.250*	57.470	*36.130*	20.560	61.234
	t_{RPE}	0.028	**0.334**	**0.033**	0.030	0.037	*0.024*	0.031	0.021	0.033	**0.033**	**0.024**	**0.057**
	r_{RPE}	*0.084*	*0.145*	*0.084*	*0.070*	0.051	*0.062*	*0.057*	*0.058*	*0.068*	*0.072*	*0.073*	0.075

Note: Top three rankings are shown in bold, underlined and italic text respectively.



(a) sequence 07 (b) sequence 09 (c) sequence 10

(d) sequence 01 (e) sequence 05 (f) sequence 08

FIGURE 4. Estimated trajectories of Our (red), DF-VO (blue), ORB-SLAM2 (green) and Depth-VO-Feat (magenta) in different sequences of the KITTI dataset. The ground truth (black dotted) is also plotted. The starting point is shown with a black square box.

by 0.017 [m] in translation and by 0.006 [deg] in rotation. Overall, the HardNet attains the top position in terms of $t_{err}(\%)$ and $r_{err}(\%)$ while it is a runner-up in terms of ATE, t_{RPE} and r_{RPE}.

Next, we present a comparison of our method (with HardNet as the deep descriptor) with five state-of-the-art methods namely ORB-SLAM2 [5], convLSTM [27], Deep-VO-Feat [11], DF-VO [13] and SF-VO [28] in Table. 3. The ORB-SLAM2 is a widely-used geometry-based method employing ORB descriptor; convLSTM is a deep learning-based method employing a self-learning mechanism for domain adaptation; Deep-VO-Feat, DF-VO and SF-VO are hybrid methods like ours combining deep learning and geometry-based methods. Moreover, ORB-SLAM2, convLSTM, Deep-VO-Feat and SF-VO deal with monocular VO while DF-VO exploits stereo information. We have used the results reported in [13] for ORB-SLAM2 and Deep-VO-Feat in addition to their DF-VO results. It is noteworthy that Deep-VO-Feat uses the KITTI dataset for training a depth network. Similarly, DF-VO uses the dataset for dense optical flow and depth networks. Thus, both Deep-VO-Feat and DF-VO are adapted to the driving sequences. Contrary to these methods, SF-VO uses synthetic datasets for learning sparse optical flow. The ORB-SLAM2 without loop closure is used for comparison in this work. The results of convLSTM trained on the synthetic dataset[1] are reported here for a fair comparison. However, only $t_{err}(\%)$ and $r_{err}(\%)$ are reported by the authors of convLSTM in [27]. In [28], the evaluation of SF-VO is demonstrated using $t_{err}(\%)$, $r_{err}(\%)$ and ATE only. It is also important to note that monocular VO methods lack the metric scale, thus the ground truth is used to estimate the proper scale as proposed in [13]. In Table 3, the top three methods for each sequence and each metric are shown in bold, underlined and italic text respectively. In terms of average errors computed on all sequences, SF-VO achieves the lowest drift in the translation compared to all other methods while our method is ranked second by a narrow margin. Specifically, SF-VO obtains 3.185% as t_{err} while our method lags by just 0.426%. Both SF-VO and our method perform significantly better than DF-VO (6.747%), Depth-VO-Feat (9.354%), ORB-SLAM2 (17.119%) and convLSTM (17.695%) in terms of drift in the translation. However, our method obtains the smallest relative translation error (t_{RPE}) of 0.057 [m] compared to the runner-up (DF-VO) obtaining 0.138 [m] while Depth-VO-Feat and ORB-SLAM2 obtain 0.148 [m] and 0.388 [m] respectively. The ORB-SLAM2 attains the lowest drift in the rotation (r_{err} = 0.357%, r_{RPE} = 0.066 [deg]) and our method holds the second place by a narrow margin (r_{err} = 1.186%, r_{RPE} = 0.057 [deg]). SF-VO obtains the third place in the ranking with an r_{err} of 1.484% while DF-VO is ranked third in terms of r_{RPE} (0.105 [deg]). The convLSTM has significantly higher r_{err} (4.676%) while Depth-VO-Feat performs slightly better in comparison (r_{err} = 3.314%, r_{RPE} = 0.187 [deg]). In terms of ATE, SF-VO, Deep-VO-Feat and Deep-VO are the top three methods obtaining 22.095 [m], 48.695 [m] and 49.855 [m] respectively. Our method and ORB-SLAM2 are ranked fourth and fifth in terms of ATE obtaining 61.234 [m] and 69.727 [m] respectively.

In addition to quantitative results, trajectories estimated by ORB-SLAM2, Deep-VO-Feat, DF-VO and our method are plotted against the ground truth in several sequences of the KITTI dataset are presented in Fig. 4. Unfortunately, the estimated trajectories of SF-VO and convLSTM are not available publicly and therefore, we could not include them in our qualitative analysis. It can be observed that our method performed well in sequences 07, 09 and 10 shown in the first row of the figure. The second row displays sequences where our method deviates from the ground truth significantly. In sequence 01 (Fig. 4(d)), all methods including ours could not perform well but our method tracked the camera motion long enough compared to all other methods. In sequences 05 (Fig. 4(e)) and 08 (Fig. 4(f)), DF-VO outperformed all other methods including ours.

VI. CONCLUSION AND FUTURE WORKS

In this article, we have presented a geometry-based VO method for stereo camera setup. The performance of the geometry-based VO method relies on the accurate tracking of points between the camera views. Generally, the method employs a descriptor-matching technique to establish correspondences between different camera views which requires robust descriptors. In this work, we have shown through experiments that deep descriptors, owing to the recent developments in the domain of deep learning, perform better than their conventional counterparts on the KITTI benchmark dataset which presents a challenging real-world scenario. It is noteworthy that deep descriptors are not trained on the KITTI dataset in this work and their performance relies on the generalization capability of networks for descriptor learning. Comparison with state-of-the-art methods demonstrates the effectiveness of our method. In comparison, it achieves competitive relative and absolute pose errors.

In future works, we aim to investigate the performance of deep descriptors by training networks on the KITTI dataset and compare their performance with the pre-trained ones. We also plan to consider multiple datasets presenting different scenarios (indoor/outdoor, ground/aerial vehicles, etc.) in our evaluation. Furthermore, we will focus on the joint learning framework to learn of keypoints and descriptors in an image in an end-to-end manner, especially for VO/SLAM pipelines.

ACKNOWLEDGMENT

The authors would like to thank the Deanship of Scientific Research (DSR), King Abdulaziz University (KAU), Jeddah, Saudi Arabia for their financial support. *(Muhammad Bilal, Muhammad Shehzad Hanif, Khalid Munawar, and Ubaid M. Al-Saggaf contributed equally to this work.)*

[1]Carla simulator: www.carla.org

REFERENCES

[1] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robot. Autom. Mag.*, vol. 18, no. 4, pp. 80–92, Dec. 2011.

[2] K. Wang, S. Ma, J. Chen, F. Ren, and J. Lu, "Approaches, challenges, and applications for deep visual odometry: Toward complicated and emerging areas," *IEEE Trans. Cognit. Develop. Syst.*, vol. 14, no. 1, pp. 35–49, Mar. 2022.

[3] S. Hwang, M. Cho, Y. Ban, and K. Lee, "Frame-to-frame visual odometry estimation network with error relaxation method," *IEEE Access*, vol. 10, pp. 109994–110002, 2022.

[4] X. Gao and T. Zhang, *Introduction to Visual Slam: From Theory to Practice*. Cham, Switzerland: Springer, 2021.

[5] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[6] I. Cvišic and I. Petrovic, "Stereo odometry based on careful feature selection and tracking," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Sep. 2015, pp. 1–6.

[7] M. Persson, T. Piccini, M. Felsberg, and R. Mester, "Robust stereo visual odometry from monocular techniques," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 686–691.

[8] T. Pire, T. Fischer, G. Castro, P. D. Cristóforis, J. Civera, and J. J. Berlles, "S-PTAM: Stereo parallel tracking and mapping," *Robot. Auto. Syst.*, vol. 93, pp. 27–42, Jul. 2017.

[9] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[10] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 225–234.

[11] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. M. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 340–349.

[12] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 2043–2050.

[13] H. Zhan, C. S. Weerasekera, J. Bian, and I. Reid, "Visual odometry revisited: What should be learnt?" in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 4203–4210.

[14] C. Zhao, Y. Tang, Q. Sun, and A. V. Vasilakos, "Deep direct visual odometry," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7733–7742, Jul. 2022.

[15] X. Ban, H. Wang, T. Chen, Y. Wang, and Y. Xiao, "Monocular visual odometry based on depth and optical flow using deep learning," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–19, 2021.

[16] S. Zagoruyko and N. Komodakis, "Deep compare: A study on using convolutional neural networks to compare image patches," *Comput. Vis. Image Understand.*, vol. 164, pp. 38–55, Nov. 2017.

[17] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan, "Image matching from handcrafted to deep features: A survey," *Int. J. Comput. Vis.*, vol. 129, no. 1, pp. 23–79, Jan. 2021.

[18] M. Brown, G. Hua, and S. Winder, "Discriminative learning of local image descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 43–57, Jan. 2011.

[19] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "HPatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3852–3861.

[20] B. G. V. Kumar, G. Carneiro, and I. Reid, "Learning local image descriptors with deep Siamese and triplet convolutional networks by minimizing global loss functions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5385–5394.

[21] M. S. Hanif, "Patch match networks: Improved two-channel and Siamese networks for image patch matching," *Pattern Recognit. Lett.*, vol. 120, pp. 54–61, Apr. 2019.

[22] Y. Tian, B. Fan, and F. Wu, "L2-Net: Deep learning of discriminative patch descriptor in Euclidean space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6128–6136.

[23] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4829–4840.

[24] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Sep. 1999, pp. 1150–1157.

[25] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[26] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[27] S. Li, X. Wang, Y. Cao, F. Xue, Z. Yan, and H. Zha, "Self-supervised deep visual odometry with online adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6338–6347.

[28] Q. Liu and B. Chen, "Robust visual odometry using sparse optical flow network," *Eng. Appl. Artif. Intell.*, vol. 116, Nov. 2022, Art. no. 105471.

[29] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part II: Matching, robustness, optimization, and applications," *IEEE Robot. Autom. Mag.*, vol. 19, no. 2, pp. 78–90, Jun. 2012.

[30] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

[31] A. Alapetite, Z. Wang, J. P. Hansen, M. Zajączkowski, and M. Patalan, "Comparison of three off-the-shelf visual odometry systems," *Robotics*, vol. 9, no. 3, p. 56, Jul. 2020.

[32] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 3946–3952.

[33] S. Song, M. Chandraker, and C. C. Guest, "Parallel, real-time monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 4698–4705.

[34] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—A modern synthesis," in *Proc. Int. Workshop Vis. Algorithms*. Cham, Switzerland: Springer, 2000, pp. 298–372.

[35] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3D reconstruction in real-time," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 963–968.

[36] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–770, Jun. 2004.

[37] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[38] F. Steinbrücker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Nov. 2011, pp. 719–722.

[39] J. Jiao, J. Jiao, Y. Mo, W. Liu, and Z. Deng, "MagicVO: An end-to-end hybrid CNN and bi-LSTM method for monocular visual odometry," *IEEE Access*, vol. 7, pp. 94118–94127, 2019.

[40] T. Pandey, D. Pena, J. Byrne, and D. Moloney, "Leveraging deep learning for visual odometry using optical flow," *Sensors*, vol. 21, no. 4, p. 1313, Feb. 2021.

[41] F. Xue, X. Wang, J. Wang, and H. Zha, "Deep visual odometry with adaptive memory," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 940–954, Feb. 2022.

[42] R. Zhu, M. Yang, W. Liu, R. Song, B. Yan, and Z. Xiao, "DeepAVO: Efficient pose refining with feature distilling for deep visual odometry," *Neurocomputing*, vol. 467, pp. 22–35, Jan. 2022.

[43] Z. Liu, E. Malis, and P. Martinet, "A new dense hybrid stereo visual odometry approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 6998–7003.

[44] Y.-J. Cao, X.-S. Zhang, F.-Y. Luo, P. Peng, C. Lin, K.-F. Yang, and Y.-J. Li, "Learning generalized visual odometry using position-aware optical flow and geometric bundle adjustment," *Pattern Recognit.*, vol. 136, Apr. 2023, Art. no. 109262.

[45] R. Spangenberg, T. Langner, and R. Rojas, "Weighted semi-global matching and center-symmetric census transform for robust driver assistance," in *Computer Analysis of Images and Patterns*. Cham, Switzerland: Springer, 2013, pp. 34–41.

[46] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision With the OpenCV Library*. Sebastopol, CA, USA: O'Reilly Media, 2008.

[47] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 930–943, Aug. 2003.

[48] S. Gupta, M. Kumar, and A. Garg, "Improved object recognition results using SIFT and ORB feature detector," *Multimedia Tools Appl.*, vol. 78, no. 23, pp. 34157–34171, Dec. 2019.

[49] D.-M. Jeong, J.-H. Kim, Y.-W. Lee, and B.-G. Kim, "Robust weighted keypoint matching algorithm for image retrieval," in *Proc. 2nd Int. Conf. Video Image Process.*, Dec. 2018, pp. 145–149.
[50] Z. Yan, H. Wang, Q. Ning, and Y. Lu, "Robust image matching based on image feature and depth information fusion," *Machines*, vol. 10, no. 6, p. 456, Jun. 2022.
[51] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[52] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 835–846, 2006.
[53] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.
[54] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *Int. J. Comput. Vis.*, vol. 74, no. 1, pp. 59–73, Aug. 2007.
[55] M. Bansal, M. Kumar, and M. Kumar, "2D object recognition: A comparative analysis of SIFT, SURF and ORB feature descriptors," *Multimedia Tools Appl.*, vol. 80, no. 12, pp. 18839–18857, May 2021.
[56] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2006, pp. 430–443.
[57] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 778–792.
[58] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016, p. 119.
[59] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *Proc. 23rd ACM Int. Conf. Multimedia*, Oct. 2015, pp. 689–692.
[60] M. Abadi, "*TensorFlow*: A system for *Large−Scale* machine learning," in *Proc. USENIX Symp. Oper. Syst. Design Implement.*, 2016, pp. 265–283.
[61] A. Paszke, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8026–8037.

MUHAMMAD BILAL was a Postdoctoral Researcher with KAIST, South Korea. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, King Abdulaziz University (KAU). He is also an educator, a researcher, and a maker. His research interests include digital image/signal processing, machine learning/AI, digital/analog circuit designs, embedded systems, and robotics.

MUHAMMAD SHEHZAD HANIF received the B.Sc. degree in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan, in 2001, and the M.S. degree in engineering sciences and the Ph.D. degree in computer engineering from Sorbonne University, Paris, France, in 2006 and 2009, respectively. He is currently an Associate Professor with the Department of Electrical and Computer, King Abdulaziz University, Jeddah, Saudi Arabia. His research interests include machine learning, image analysis, and information fusion.

KHALID MUNAWAR received the B.S. degree in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan, in 1992, and the M.S. and Ph.D. degrees in aeronautics and space engineering from Tohoku University, Sendai, Japan, in 1997 and 2000, respectively. He is currently a Professor of electrical and computer engineering with King Abdulaziz University, Jeddah, Saudi Arabia. He was an Assistant Professor with Tohoku University, and an Assistant and Associate Professor with the National University of Sciences and Technology, Islamabad, Pakistan, before moving to King Abdulaziz University, in 2012. His current research interests include control systems, industrial automation, unmanned and autonomous systems, and navigation and control.

UBAID M. AL-SAGGAF (Member, IEEE) received the B.Sc. degree (Hons.) in electrical engineering and mathematics from the King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia, in 1980, and the M.Sc. and Ph.D. degrees in electrical engineering from Stanford University, USA, in 1983 and 1986, respectively. He joined KFUPM, as an Assistant Professor, in 1986. From 1992 to 2010, he was on leave working for the Ministry of Defense and Aviation as an Executive Technical Advisor with the Research and Development Department. He joined King Abdulaziz University (KAU), in September 2010, where he is currently a Professor. He is also the Founder and the Director of the Center of Excellence in Intelligent Engineering Systems (CEIES) and the Founder of the Innovation and Prototyping Center (IPC), KAU. His research interests include broad spectrum from theoretical to practical aspects of engineering, including systems, control, communications, and signal processing.