International Journal on Advanced Science Engineering Information Technology

Vol.8 (2018) No. 4-2
ISSN: 2088-5334

Multiple Descriptors for Visual Odometry Trajectory Estimation

Mohammed Salameh[*1], Azizi Abdullah[#2], Shahnorbanun Sahran[#3]

Center for Artificial Intelligence Technology
Faculty of Information Science and Technology
University Kebangsaan Malaysia, 43600 Bangi, Malaysia
*[*1]m.omar82@siswa.ukm.edu.my; [#2]azizia@ukm.edu.my; [#3]shahnorbanun@ukm.edu.my*

***Abstract*— Visual Simultaneous Localization and Mapping (VSLAM) systems are widely used in mobile robots for autonomous navigation. One important part of VSLAM is trajectory estimation. Trajectory estimation is a part of the localization task in VSLAM where a robot needs to estimate the camera pose in order to precisely align the real visited image locations. The poses are estimated using Visual Odometer Trajectory Estimation (VOTE) by extracting distinctive and traceable key points from sequence image locations having been visited by a robot. In the visual trajectory estimation, one of the most popular solutions is arguably PnP-RANSCA function. PnP-RANSAC is a common approach used for estimating the VOTE, which uses a feature descriptor such as SURF to extract key-points and match them in pairs based on their descriptors. However, due to the sensor noise and the high fluctuating scenes constitute an inevitable shortcoming that reduces the single visual descriptor performance in extracting the distinctive and traceable key points. Thus, this paper proposes a method that uses a random sampling scheme to combine the result of multiple key-points descriptors. The scheme extracts the best key points from SIFT, SURF and ORB key-point detectors based on their key-point response value. These key points are combined and refined based on Euclidean distances. This combination of key points with their corresponding visual descriptors is used in VOTE, which reduces the trajectory estimation errors. The proposed algorithm is evaluated on the widely used benchmark dataset KITTI where the three longest sequences are selected, 00 with 4541 images, 02 with 2761 images, and 05 with 1101 images. In trajectory estimation experiment, the proposed algorithm can reduce the trajectory error of 44%, 8% and 13% on KITTI dataset for the sequence 00, 02 and 05 respectively based on translational and rotational errors. In addition, the proposed algorithm succeeded in reducing the number of key points used in VOTE as combined with the state-of-the-art RTAB-Map.**

***Keywords*—Visual Odometer; trajectory estimation; structure from motion; RANSAC; selection scheme; feature matching**

I. INTRODUCTION

Accurate Visual Odometry (VO) is one of the crucial elements in Visual Simultaneous Localization and Mapping (VSLAM) for autonomous navigation and driving assistance based on computer vision. VO can be identified as a process of estimation of a camera pose and motion between a sequence of image locations where the Trajectory Estimation (TE) is a process, which takes camera poses and enriches them with time information. Nister et al. proposed the Visual odometer (VO) term in 2004 [22]. National Aeronautics and Space Administration (NASA) used the VO in their rovers for Mars exploration missions [18]. Since that time, the VO has occupied a large and a growing share in the field of robotics and autonomous navigation even with computer vision application such as Content-Based Image Retrieval (CBIR).

In the literature, the pose estimation problem is known as Perspective-n-Point (PnP) which determines the pose of a camera based on the apparent position of number n key points extracted from image locations [7]. There are many methods proposed for Visual Odometry Trajectory Estimation (VOTE) based on a PnP problem such as Direct Linear Transform (DLT) or Random Sample Consensus (RANSAC) [6]. However, the DLT method might produce solutions, which are not valid in particular orthogonal due to estimation step.

Perspective-n-Point using RANSAC (PnP-RANSAC) scheme has been used to estimate the pose which is used for constructing the trajectory of a robot. The progress of RANSAC has earned considerable attention from researchers since Fischler, and Bolles [6] introduced this method regarding the convergent speed and performance [11], [16], [30]. One of the problems in the RANSAC is profoundly affected by the number and the distribution of the key points. RANSAC picks up the sample randomly, the number of key points affects the degree of variation of outputs to the same inputs, and the distribution of key points is a significant factor for efficient trajectory estimation [13], [24], [31].

Different approaches tackle the problem of critical points distribution by dividing the image location into sub regions overlapping or non-overlapping in order to decrease the number, and the distribution of key points [2]. However, dividing the image locations into sections is an undesirable way in configuration [13], [26], [31]. The number of points and their distribution associated with the correct matched Key points can improve the VO performance [28]. However, VOTE approaches which using PnP-RANSAC scheme with a single Key point's detector cannot extract the suitable number and distribution of the Key points.

The key point's detection method needs to be robust and be able to find similar key points in the previous images regardless of any differences in image scaling, rotation, or variance of illumination. Key point's detection has a primary influence on the accuracy of estimating the calibration matrix and the fundamental matrix, which are used for estimating the camera poses [27], [26].

Therefore, this research focuses on the key point's detection stage in VOTE and proposes a new selection scheme for enhancing a stereo VO framework named Multiple Descriptors for Visual Odometry Trajectory Estimation (MD-VOTE). The proposed framework extracts and selects the most responsive key points using three visual descriptors named Speeded-Up Robust Features (SURF), Scale-Invariant Feature Transform (SIFT) and Oriented FAST and Rotated BRIEF (ORB). These key points are refined and combined in order to keep the distribution, and the key points with the minimum number of key points, which improves the RANSAC performance and estimates a high accurate trajectory of a robot.

Real-Time Appearance-Based Mapping (RTAB-Map) is the state-of-the-art VSLAM approach, and it is a single visual descriptor approach, which can estimate a robot trajectory using PnP-RANSAC 3D-2D method. RTAB-Map is used as a based method to compare with the proposed MD-VOTE. The proposed algorithm MD-VOTE is evaluated on the outdoor dataset Vision Benchmark Suite from Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI), and the evaluation results are compared with RTAB-Map. The proposed MD-VOTE significantly outperforms RTAB-Map regarding translational and rotational errors on the longest three sequences 00, 02, and 05 from KITTI datasets.

This paper is organized as follows: Section (II) highlights a general VOTE framework and its challenges in extracting the distinctive key points and describes the proposed MD-VOTE algorithm. Section (III) presents the experimental setup, and the results of the trajectory estimation obtained by the proposed MD-VOTE algorithm and RTAB-Map examined under different conditions on the KITTI datasets. Finally, Section (IV) gives a summary of the paper.

II. Material and Method

A new selection scheme for enhancing VOTE named Multiple Descriptors for Visual Odometry Trajectory Estimation (MD-VOTE) is proposed to use the multiple descriptors "SURF, SIFT and ORB" based on the PnP-RANSAC scheme to estimate the robot's trajectory along visited locations. The proposed algorithm selects a set of distinctive 3*D*-2*D* matching key points, which are extracted from the image locations by using the key points detectors method of each descriptor individually. Algorithm 1 illustrates the proposed MD-VOTE procedures and Figure 1 shows the flowchart of the proposed MD-VOTE algorithm.

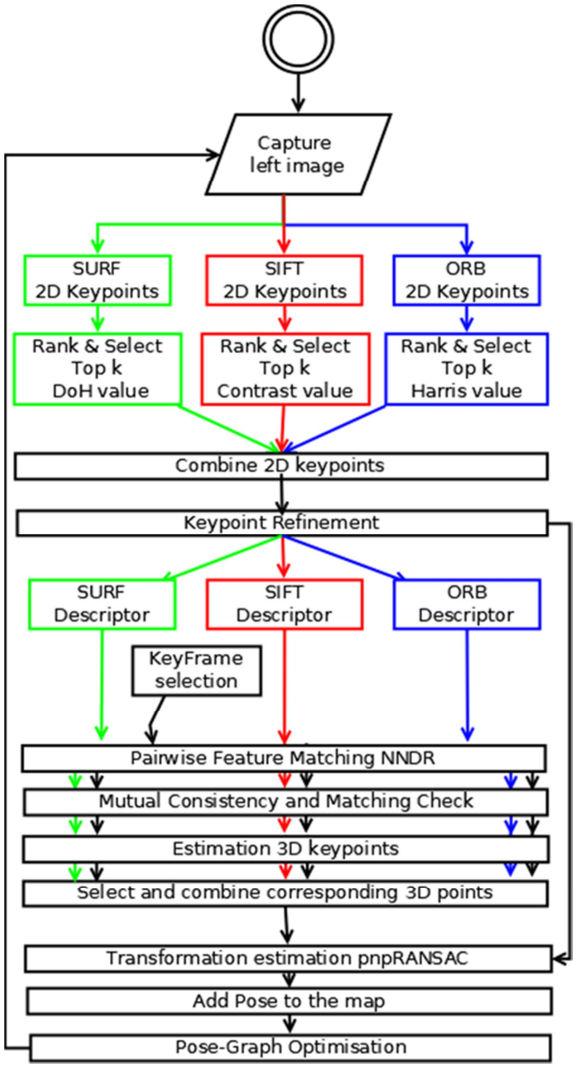


Fig. 1: The flowchart for the proposed MD-VOTE algorithm

A. *Key points Detection Stage*

At the first stage, the key points detector from each visual descriptor: SURF, SIFT, and ORB detect their key points individually from the current image location. After that, the proposed algorithm ranks each key points according to its response value, followed by the refinement process. Next, the visual features are extracted from each refined key point, and the description of such key points is generated based on its corresponding descriptor.

The final process in this stage is the key frame selection where the proposed algorithm used the method of RTAB-Map in selecting the key frame according to a pre-set threshold. The same key frame will remain until the number of inliers points become under the threshold [15].

1) Ranking and Selecting Key points: The proposed algorithm starts by detecting three sets of key points using the three visual descriptors SURF, SIFT, and ORB from the current image location. Each set of key points is ranked individually based on the key points response value. The SIFT's key points detector applies the Differences of

Gaussian (DoG) in order to recognize as much as possible of the key points by generating several Gaussian-blurred images. These images are based on several scales of the input image.

After that, the SIFT's key points detector computes the DoG images based on the subtraction of neighbors in scale space from each other. Based on the DoG images, the key points are selected if they meet the following conditions: (1) they are locally extreme in the DoG images in space and scale. (2) They fulfill the threshold ratio of eigenvalues of the Hessian matrix. (3) The key points contrast is high. The key points which succeed are detected by interpolating through the DoG images [17]. The contrast value is the key points response value, which is used to determine how strong the key points are [12].

The SURF descriptor is partly inspired by SIFT, where the key points in SURF start with computing integral in images which are fast in generating the Laplacian of Gaussian images using a box filter with various sizes. After that, the key points are detected as local maxima of the DoH on different levels applied to the integral image [1]. Since the key points are selected and extracted based on the DoH value [1], the DoH value is also used as the key points response value which is used to determine the strength of the key points [12].

The ORB develops Orientation FAST Key point (OFAST) for the key points detector which enhances the Features for Accelerated Segment Test (FAST) detector. The OFAST detects the key points from the input image based on the FAST detector with the radius of 9 for the circular of the connected pixels around the corner. Then, the key points are sorted out based on Harris corner computations to select the top key points [25]. The Harris corner computation produces the key point's response value, which is used to determine how strong the key points are [12].

Algorithm 1 : MD-VOTE

1: kp_{SURF} is a set of 2D keypoints detected by SURF keypoints detector from the current image I_t
2: kp_{SIFT} is a set of 2D keypoints detected by SIFT keypoints detector from the current image I_t
3: kp_{ORB} is a set of 2D keypoints detected by ORB keypoints detector from the current image I_t
4: Ranked the keypoints kp_{SURF} based on the Determinant of Hessian (DoH) value of each keypoint.
5: Ranked the keypoints kp_{SIFT} based on the contrast value of each keypoint.
6: Ranked the keypoints kp_{ORB} based on the Harris corner measure value of each keypoint.
7: top_{SURF} is a set of the top k keypoints from kp_{SURF}
8: top_{SIFT} is a set of the top k keypoints from kp_{SIFT}
9: top_{ORB} is a set of the top k keypoints from kp_{ORB}
10: kp_{all} is a set of the combination of keypoints
$$kp_{all} = top_{SURF} \cup \text{top}_{\text{SIFT}} \cup \text{top}_{\text{ORB}}$$
11: $kp_{refined} = KeypointRefinement(kp_{all}, d_{min})$
12: $f = FeatureExtraction\begin{pmatrix} kp_{refined}, SURF, SIFT \\ , ORB \end{pmatrix}$ extracts the corresponding features of each keypoints based on its own descriptor.
13: $kp_{pair} = PairwiseMatching(kp_{refined}, f, I_{keyframe})$, where $I_{keyframe}$ is the keyframe image.
14: kp_{3D} is a set of 3D keypoints extracted from a keyframe image based on the corresponding kp_{pair}.
15: $pose_t$ is the corresponding pose of the camera at time t which is estimated by using $PnP - RANSAC(kp_{refined}, kp_{3D})$.

After the ranking of each key points set, the top k key points are selected from each key points set and are combined using Equation 1

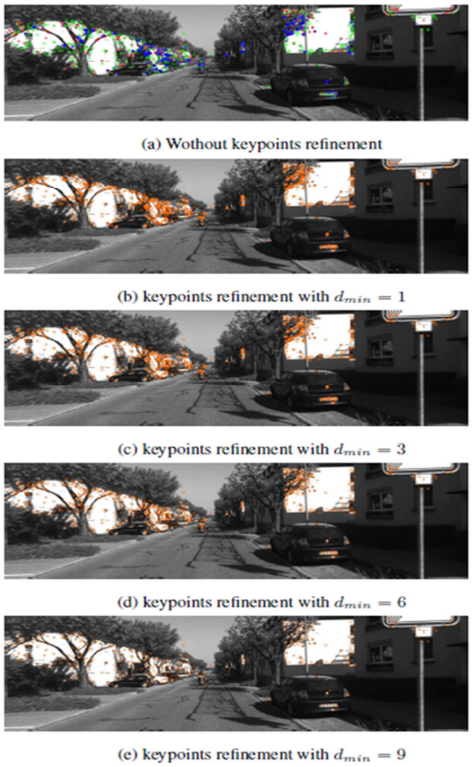
$$\text{kp}_{\text{all}} = \text{top}_{\text{SURF}} \cup \text{top}_{\text{SIFT}} \cup \text{top}_{\text{ORB}} \quad (1)$$

where top_{SURF}, top_{SIFT}, top_{ORB} are the sets of the top k key points extracted from each descriptor, the value of the variable "top k" was selected as the top 400 key points based on [14] work and [is the union operation as Equation 2

$$\text{A} \cup \text{B} = \{\text{x} \mid \text{x} \in \text{A or x} \in \text{B}\} \quad (2)$$

where *A* and *B* are two sets of key points. Each descriptor covers up the shortages of each other, and the result of this combination is a set of distinctive key points extracted from the integration of these descriptors. Finally, the set kp_{all} is passed to the next process to keep the distinctive key points and eliminate the overlapped key points as described in the next paragraph.

2) Key points Refinement: The result of the previous process is the set of key point's kp_{all}, which contains the top *k* key points detected by each descriptor. In this stage, the key point's refinement method keeps the distinctive key points and eliminates the overlapped ones. The distinctive key points are selected according to the highest response value in a limited area. The area size is defined by the radius d_{min}.


(a) Wothout keypoints refinement

(b) keypoints refinement with $d_{min} = 1$

(c) keypoints refinement with $d_{min} = 3$

(d) keypoints refinement with $d_{min} = 6$

(e) keypoints refinement with $d_{min} = 9$

Fig. 2: A simple example for the refining method.

Fig. 2 shows an example of the refining method as follows:

- Sorts out the key points in the set kp_{all} based on the key point's location in the image.
- kp_d is a set of key points which includes the first key points kp_0 in the set kp_{all}.
- The set kp_d also includes all key points lying within a distance d_{min} from the key points kp_0.
- The function selects the key points from the set kp_d, which has the highest rank based on its corresponding descriptor. As shown in Fig. 2, the set kp_d includes the key points [kp_1, kp_2, kp_3] whose corresponding ranks are [2], [5], [9] respectively. The key point's *kp2* is selected because it obtained the highest rank, as the key points with the highest rank is more distinctive and traceable.
- The selected key points from the previous step is added to the set kp_{refine} and the set kp_d is eliminated from the set kpall.
- The steps (2,3,4,5) are repeated until kp_{all} is empty.

Fig. 3: Example for the key points refinement method using a different radius

Fig. 3 depicts an example of the number of key points detected by the three visual descriptors SURF, SIFT and ORB
and shows the impact of key points refinement method using a different radius on the number of key points. The image location in the example is the location number 5 taken from sequence 05 of the KITTI dataset. This example shows that the distribution of the key points is focused on the edges of the white wall and the car plate. The proposed refinement method tries to find the suitable distribution of the key points, which improves the trajectory estimation.

B. Key points Tracking Stage

The second stage handles the pairwise matching of the key points and checks the mutual consistency. After refining the combined key points, each descriptor processes the pairwise matching of its keypoints. This set of key points is pairwise matched between the current image location and the key frame image location using the Nearest Neighbor Distance Ratio (NNDR) approach with kd tree and produces the set of matched pair-key points [14]. Then, the proposed algorithm extracts the 3D points from the key frame based on the corresponding 2D key points. At this stage, the refined 2D key points and their corresponding 3D key points are passed on to the next stage of motion estimation.

C. Motion Estimation Stage

The motion estimation stage is the stage where the pose is estimated between the current image location and the key frame location. The refined 2D key points and their corresponding 3D key points are used to estimate the translational and rotational matrix by using PnP-RANSAC approach, which eliminates the outlier. Finally, the pose of the camera related to the current image location is saved in the map representing the movement motion between the two locations as translational and rotational matrices.

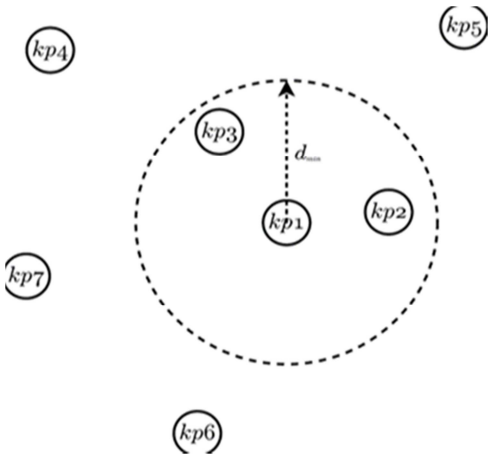


Fig. 4: Samples of KITTI dataset

Regarding the graph optimization, the proposed algorithm used the Tree-based netwORk Optimizer (TORO) [10] that is being used in RTAB-Map [15].

III. Results and Discussion

A. Experimental Setup

The proposed MD-VOTE algorithm performance was evaluated using three outdoor scenes: the sequences 00, 02, and 05 from the public dataset KITTI [9]. KITTI is a stereo-image dataset provided with visual odometry ground truth for the sequences. This dataset is used extensively in the literature to evaluate the performance of trajectory estimation algorithms [5, 21, 20, 23, 4]. Table 1 summarises the KITTI sequences and Fig. 4 shows samples of KITTI dataset.

TABLE I
THE KITTI SEQUENCES DATASETS [9].

Dataset	#Images	Image size (px)	Dist(Km)
KITTI 00	4541	1241x370	3.73
KITTI 05	2761	1226x370	2.22
KITTI 07	1101	1226x370	1.27

The average translational and average rotational error criteria are used to evaluate the performance of the proposed MD-VOTE where the translational error is measured in percentages, and the rotational error is measured in degrees

per meter. KITTI Benchmark Suite is a toolbox to assess the trajectory error. The trajectory error estimates the relative average of the translational error Etrans and rotational error Erot using the segments at 100m, 200m, …, 800m lengths. It was conducted on the 3D coordinates of the estimated trajectory and the ground truth as follows [19].

$$E_{trans}(F) = \frac{1}{|F|} \sum_{(i,j)\in F} \left|\left|(\hat{p}_j \ominus \hat{p}_i) \ominus (p_j \ominus p_i)\right|\right|_2 \quad (3)$$

$$E_{rot}(F) = \frac{1}{|F|} \sum_{(i,j)\in F} \angle\left[(\hat{p}_j \ominus \hat{p}_i) \ominus (p_j \ominus p_i)\right] \quad (4)$$

where F is a set of frames (i, j), $\hat{p} \in SE(3)$ is the estimate poses and $p \in SE(3)$ is the true poses and $\angle[.]$ is the rotation angle. The SE(3) is a special Euclidean group, refer to transformations matrix and the $\ominus$ stand for the inverse compositional operator [8], [9]. The relative change is used to measure the difference between the two values where one of the values is considered as reference "initial" values xi and relative change is unit less percentages expressed the percentage change between the initial value and the final value using Equation 5 [29]

$$C_r(x_f, x_i) = \frac{x_f - x_i}{|x_i|} \quad (5)$$

where $C_r$is relative change, x_i is the initial value and x_f is the final value relative change is used to measure the difference that has been made by the proposed algorithm MD-VOTE in trajectory estimation against the start-of-the-art algorithm. Relative change is used to measure the difference that has been made by the proposed algorithm MD-VOTE in trajectory estimation against the start-of-the-art algorithm. All the parameters of the visual descriptors SURF. SIFT and ORB are set as reported in OpenCV [12]. The value of the variable maximum number of key points extracted from the image being determined at 400 points for each descriptor. This value is a default set by RTAB-Map [15]. For comparative purposes, RTAB-Map is used as baseline VOTE method to compare with the proposed MD-VOTE algorithm, where MD-VOTE and RTAB-Map parameters are set up as reported in [15].

B. Evaluation

The first experiment evaluates the performance of the benchmark VOTE using the three visual descriptors SURF, SIFT and ORB individually whereas each descriptor is tested with a different number of key points: 400 key points and 1000 key points. This experiment is conducted on the same three sequences (00, 02, and 05) in the KITTI dataset. The experiments are divided into three categories for simplicity in the evaluation and presentation of the results.

1) Selecting the number of key points: Table 2 shows the results of the first experiment conducted using 400 and 1000 key points in trajectory estimation through two values, the translational and rotational errors. An average translational error is measured in percentages, and an average rotational error is measured in degrees per meter.

The results show that the benchmark VOTE using the visual descriptor ORB has scored the least errors compared to SURF and SIFT using the three sequences with a maximum of 1000 extracted key points used in trajectory estimation. In contrast, with a maximum of 400 extracted key points, the benchmark VOTE using the visual descriptor SURF has scored the least errors compared to SIFT and ORB using the same sequences. The reason is that SURF can extract and match the 400 key points more efficiently than other feature descriptors, which improve the PnP-RANSAC process in estimating the poses.

TABLE II
COMPARISON BETWEEN DIFFERENT VISUAL DESCRIPTORS USING A DIFFERENT NUMBER OF KEY POINTS

		Seq-00, 4541 Images		Seq-02, 4661 Images		Seq-05, 2761 Images	
		Translation error %	Rotation error d/m	Translation error %	Rotation error d/m	Translation error %	Rotation error d/m
400 Keypoints	SURF	0.016375	1.13E-04	0.013978	7.40E-05	0.010763	6.50E-05
	SIFT	0.019614	1.41E-04	0.017221	9.90E-05	0.01249	6.80E-05
	ORB	0.017698	1.16E-04	0.015105	9.60E-05	0.011001	6.40E-05
1000 Keypoints	SURF	0.025192	1.38E-04	0.015103	8.38E-05	0.009958	5.60E-05
	SIFT	0.025316	1.52E-04	0.01533	9.93E-05	0.01165	7.00E-05
	ORB	0.02509	1.37E-04	0.015001	8.20E-05	0.009754	5.40E-05

Based on Table 2, the SURF with 400 key points has scored the best results in the trajectory estimation for the sequences 00 and 02 over the other descriptors even with a different number of extracted key points. This fact shows that the quality of key points is more significant than the quantity in estimating the trajectory for the long sequences such as 00 and 02. In the case of sequence 05, ORB with 1000 key points score the least errors due to its high speed in detecting and extracting the key points.

2) The refinement method evaluation: The second experiment is conducted to evaluate the performance of the proposed MD-VOTE algorithm with the value of a different radius dmin, and shows the impact of a different radius on the efficiency of the proposed algorithm in estimating the trajectory.

Table 3 shows the results of the proposed MD-VOTE algorithm using different radius $d_m = [0,1,3,6,9]$ to the proposed refinement method. This experiment shows the

influence of the key point's distribution on the estimation of trajectory using sequence 05 from KITTI dataset. The results show that the proposed MD-VOTE algorithm with radius = 1 achieved minimal errors, and based on this result the radius = 1 is adopted in all experiments.

TABLE III
TRAJECTORY ESTIMATION ERRORS FOR SEQUENCE 05-USING MDVOTE

Distance in pixel	No. Keypoints	Translation %	Rotation d/m
No filter	1200	0.010084	6.10E-05
d_{min}=1	980	0.008955	5.40E-05
d_{min}=3	719	0.010589	5.90E-05
d_{min}=6	518	0.011374	6.60E-05
d_{min}=9	266	0.012149	8.20E-05

It is noticed that using the proposed MD-VOTE algorithm without refining the key points which are 1200 key points extracted from the three visual descriptors, the MD-VOTE gets 0.010084% and 6.1E-05 m/d for the average translational and average rotational errors respectively. As for MD-VOTE with the key points refinement using radius$d_{\min} = 1$, the maximum number of key points reaches 980 key points and scores 0.008955% and 5.4E-05 d/m for the average translational and average rotational errors respectively which are the least recorded errors for the sequence 05. Furthermore, when the radius value increases, the number of key points decreases and the errors rate increases.

Fig.s 5a and 5b show the translational and rotational errors as a function of the path length and the trajectory segmented at 100, 200,…, 800m lengths [9]. It is noticed that the proposed MD-VOTE algorithm used with the key points refinement using radius $d_{\min} = 1$ estimates the trajectory for sequence 05 with errors decreasing proportionately with the distance travelled which scored the least errors, i.e., 0.004638% and 0.000022 d/m for translational and rotational errors respectively.

Additionally, Fig.s 5c and 5d show the translational and rotational errors as a function of the moving speed. It is noticed that the proposed MD-VOTE algorithm with radius $d_{\min} = 1$scored the least errors over linear change speed between 6 km/h to 12 km/h. It is concluded from Fig. 5c that the translational error increases as a vehicle move faster. As a matter of fact, the vehicle speeds up when it moves in a long straight path. Conversely, Fig. 5d shows that the rotational error is inversely proportional to speed, where the rotational error appears with the rotational movement of the vehicle, and the vehicle slows down its speed as it rotates.

Now, based on the results shown in Table 3, it was decided to select the radius $d_{\min} = 1$ to be used in other experiments because MD-VOTE algorithm with radius $d_{\min} = 1$ has scored the least trajectory estimation errors for sequence 05. Accordingly, whenever the MD-VOTE algorithm is mentioned throughout this research, it means that it is the proposed algorithm that includes the key points refinement method-using radius $d_{\min} = 1$, unless it is explicitly stated otherwise.

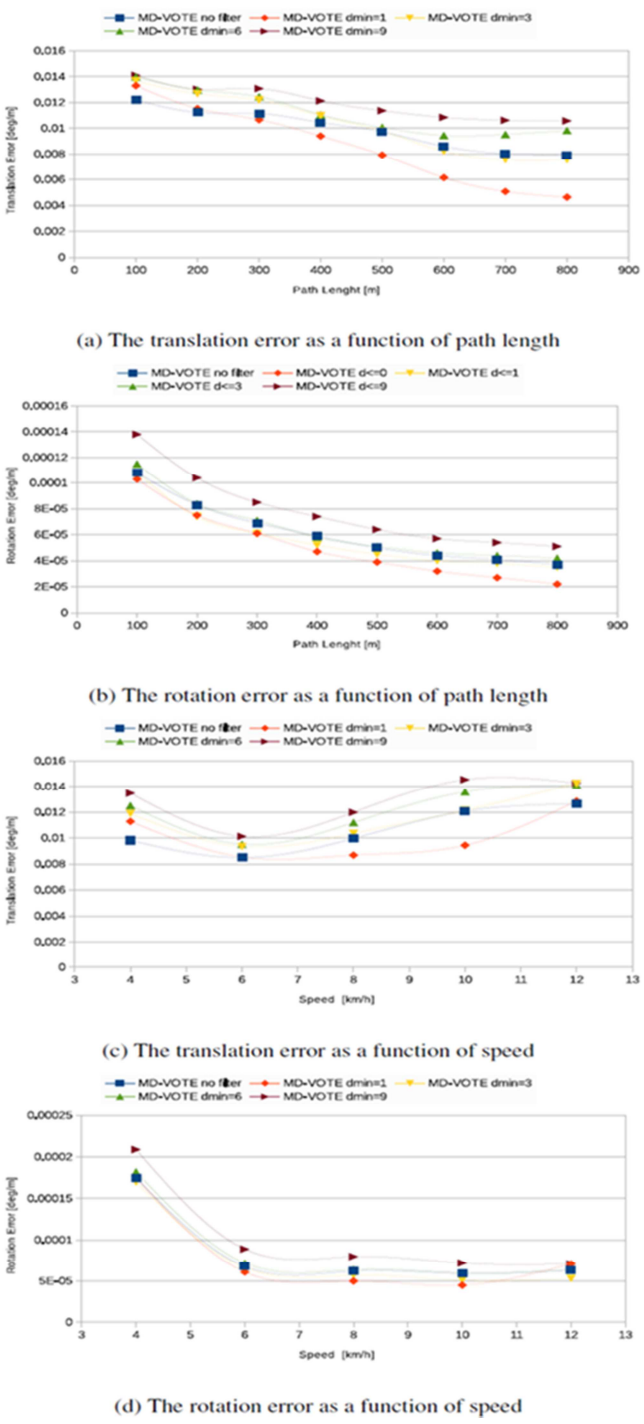


Fig. 5: Experiment on sequence 05 using MD-VOTE with different radius values

3) The MD-VOTE algorithm evaluation

The final experiment evaluates the performance of the proposed MD-VOTE algorithm and compares between the proposed MDVOTE with the key points refinement method against the standard RTAB-Map in trajectory estimation for the three sequences 00, 02 and 05 from the KITTI dataset. Table 4 shows the average translational and average rotational errors for each sequence. The last row in Table 4 shows the relative change ratio calculated between the proposed MD-VOTE algorithm with the key points refinement method and the standard RTAB-Map.

TABLE IV
TRAJECTORY ESTIMATION ERRORS FOR SEQUENCES 00, 02 AND 05 USING THE PROPOSED MD-VOTE WITH THE FILTERING METHOD AGAINST RTAB-MAP

		Translation %	Rotation d/m
Seq-00, 4541 Images	RTAB-Map	0.02%	1.26E-04
	MD-VOTE	0.01%	9.60E-05
	Relative change	-44.35%	-23.80%
Seq-02, 4661 Images	RTAB-Map	0.01%	7.70E-05
	MD-VOTE	0.01%	7.80E-05
	Relative change	-8.65%	1.29%
Seq-05, 2761 Images	RTAB-Map	0.01%	6.30E-05
	MD-VOTE	0.01%	5.40E-05
	Relative change	-13.18%	-14.28%

To clarify the efficiency of the proposed MD-VOTE algorithm, a comparison has been made between MD-VOTE and RTAB-Map regarding relative change with respect to trajectory estimation errors. The results of the comparison based on sequence 00, Table 4 show that MD-VOTE successfully reduces the translational and rotational errors by -44:35% and -23:80% respectively regarding relative changes with respect to RTB-Map. Similarly, the results of the comparison based on sequence 02, show that MD-VOTE successfully reduces the translational error by -8:65%, whereas the rotational error increases by +1:29%. As for sequence 05, both the translational and rotational errors decrease by -13:18% and -14:28% respectively.

IV. CONCLUSIONS

The PnP-RANSAC method is applied to trajectory estimation. However, the single key point's detector cannot efficiently tackle a challenging environment, which contains fluctuating scenes. Trajectory estimation requires distinctive matching key points that can be tracked to estimate the accurate trajectory of a robot's camera movement between sequences of image locations. In this paper, the proposed algorithm MD-VOTE combines the key points, which are extracted from the multiple visual descriptors, SURF, SIFT and ORB.

The combined key points are further filtered with the proposed key point's refinement method to select the most distinctive key points, which contribute to the PnP-RANSAC to improve the VOTE performance. The proposed algorithm MD-VOTE is evaluated on the longest three sequences 00, 02, and 05 from the outdoor dataset KITTI that is a widely used benchmark. The evaluation results are compared with RTAB-Map using single and multiple visual descriptors. The results of the experiments indicate that the proposed MD-VOTE significantly outperforms RTAB-Map in terms of translational and rotational errors, whereas the proposed algorithm scores the least translational and rotational errors (0.013636%, 0.000096), (0.013432%, 0.000078) and (0.008955%, 0.000054) in the three sequences 00, 02 and 05 respectively.

Additionally, the proposed MD-VOTE scores relative change ratio (-44.35%, -23.80%), (-8.65%, +1.29%) and (-13.18%, -14.28%) regarding RTAB-Map for translational and rotational errors in the three sequences 00, 02 and 05 respectively. The proposed MD-VOTE algorithm shows promising results in accurate trajectory estimation where MD-VOTE algorithm success to extract and retains the salient key points from multiple visual descriptors with suitable number and distribution for the key points. These key points improve the VO performance. As a future work, the MD-VOTE algorithm will integrate the Loop Closure Detection (LCD) to improve the VSLAM performance.

ACKNOWLEDGMENT

The authors would like to extend their appreciation and gratitude to FRGS/1/2016/ICT02/UKM/02/5 and GPK007762 grants for funding this project.

REFERENCES

[1] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). Computer vision and image understanding, 110(3):346–359, 2008.

[2] Li-Hung Chen and Kai-Wei Chiang. The performance analysis of stereo visual odometry assisted low-cost ins/gps integration system. Smart Science, 3(3):148–156, 2015.

[3] Hsiang-Jen Chien, Chen-Chi Chuang, Chia-Yen Chen, and Reinhard Klette. When to use what feature? sift, surf, orb, or a-kaze features for monocular visual odometry. In Image and Vision Computing New Zealand (IVCNZ), 2016 International Conference on, pages 1–6. IEEE, 2016.

[4] Jakob Engel, Thomas Sch¨ops, and Daniel Cremers. Lsdslam:Large-scale direct monocular slam. In Computer Vision–ECCV 2014, pages 834–849. Springer, 2014.

[5] Marco Fanfani, Fabio Bellavia, and Carlo Colombo. Accurate keyframe selection and keypoint tracking for robust visual odometry. Machine Vision and Applications, 2016.

[6] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6):381–395, 1981.

[7] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. IEEE transactions on pattern analysis and machine intelligence, 25(8):930–943, 2003.

[8] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. The International Journal of Robotics Research, 32(11):1231–1237, 2013.

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[10] Giorgio Grisetti, Slawomir Grzonka, Cyrill Stachniss, Patrick Pfaff, and Wolfram Burgard. Efficient estimation of accurate maximum likelihood maps in 3d. In Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on, pages 3472–3478. IEEE, 2007.

[11] Jie Guo, Zhihua Wei, and Duoqian Miao. Lane detection method based on improved ransac algorithm. In Autonomous Decentralized Systems (ISADS), 2015 IEEE Twelfth International Symposium on, pages 285–288. IEEE, 2015.

[12] Itseez. The OpenCV Reference Manual. Itseez, 2.4.9.0 edition, April 2014.

[13] J Kersten and V Rodehorst. Enhancement strategies for frame-to-frame uas stereo visual odometry. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, 41, 2016.

[14] Mathieu Labbe and Francois Michaud. Appearance-based loop closure detection for online large-scale and long-term operation. Robotics, IEEE Transactions on, 29(3):734–745, 2013.

[15] Mathieu Labbe and Franc¸ois Michaud. Online global loop closure detection for large-scale multi-session graph-based slam. In

Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, pages 2661–2666. IEEE, 2014.

[16] Chengbo Liu, Qiang Shen, Hai Pan, and Miao Li. Modelling and simulation: an improved ransac algorithm based on the relative angle information of samples. International Journal of Modelling, Identification and Control, 28(2):144–152, 2017.

[17] David G Lowe. Distinctive image features from scaleinvariant keypoints. International journal of computer vision, 60(2):91–110, 2004.

[18] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the mars exploration rovers. Journal of Field Robotics, 24(3):169–186, 2007.

[19] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[20] Raul Mur-Artal, JMM Montiel, and Juan D Tardos. Orbslam: a versatile and accurate monocular slam system. arXiv preprint arXiv:1502.00956, 2015.

[21] Ra´ul Mur-Artal and Juan D. Tard´os. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGBD cameras. IEEE Transactions on Robotics, 33(5):1255–1262, 2017.

[22] David Nister, Oleg Naroditsky, and James Bergen. Visual odometry. In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 1, pages I–I. Ieee, 2004.

[23] Taih´u Pire, Thomas Fischer, Javier Civera, Pablo De Crist´oforis, and Julio Jacobo Berlles. Stereo parallel tracking and mapping for robot localization. In Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pages 1373–1378. IEEE, 2015.

[24] Martin Rais, Gabriele Facciolo, Enric Meinhardt-Llopis, Jean-Michel Morel, Antoni Buades, and Bartomeu Coll. Accurate motion estimation through random sample aggregated consensus. arXiv preprint arXiv:1701.05268, 2017.

[25] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: an efficient alternative to sift or surf. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 2564–2571. IEEE, 2011.

[26] Mohammed Omar Salameh. Multiple visual descriptor Combination for loop closure detection and visual odometer trajectory estimation. Phd thesis, university kebangsaan malaysia, 2018.

[27] De-cai SHI, Xiu-cheng DONG, and Yu ZHENG. An improved orthogonal iterative algorithm for monocular camera pose estimation. DEStech Transactions on Computer Science and Engineering, 3(aics), 2016.

[28] Hauke Strasdat, JMM Montiel, and Andrew J Davison. Real-time monocular slam: Why filter? In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pages 2657–2664. IEEE, 2010.

[29] Leo T¨ornqvist, Pentti Vartia, and Yrj¨o O Vartia. How should relative changes be measured? The American Statistician, 39(1):43–46, 1985.

[30] Yue Wang, Jin Zheng, Qi-Zhi Xu, Bo Li, and Hai-Miao Hu. An improved ransac based on the scale variation homogeneity. Journal of Visual Communication and Image Representation, 40:751–764, 2016.

[31] Jun Yu, Chang-wei Luo, Chen Jiang, Rui Li, Ling-yan Li, and Zeng-fu Wang. A digital video stabilization system based on reliable sift feature matching and adaptive lowpass filtering. In CCF Chinese Conference on Computer Vision, pages 180–189. Springer, 2015.