. CORRECTING VEHICLE HEADING IN VISUAL ODOMETRY BY COMBINING KEYPOINT DETECTORS

MOHAMMED OMAR SALAMEH
AZIZI ABDULLAH
SHAHNORBANUN SAHRAN

Center for Artificial Intelligent and Technology, Universiti Kebangsaan Malaysia

ABSTRACT

This technical report describes a visual odometry algorithm for on-road vehicles under planar motion assumption. The algorithm uses as only input images taken by a single omnidirectional camera mounted on a vehicle. To achieve accurate motion estimation results, we combine a set of the-state-art-of-the-art keypoint detectors for the rotation and translation estimation. In particular, we use a set of features of different detectors based ground plane tracker to estimate the translation to estimate the rotation. As we show, the combination of strongest keypoints detectors to estimate the vehicle heading outperforms the pure single feature based approach. This pose estimation method has been successfully applied to videos from an automotive platform. We give an example of camera trajectory estimated purely from omnidirectional images over a distance of several meters.

Keywords: omnidirectional camera, visual odometry, vehicle ego-motion estimation, homography, SIFT features

1.0 INTRODUCTION

Autonomous visual robot navigation can be defined as the capability of a robot to plan an efficient path between two or more locations and execute that plan to reach the destination(s) on its own, based on visual sensors, without human intervention. A robot's efficient path and safe travel are achieved by an accurate map which represents the real surrounding environment that includes the poses of locations that a robot travelled through. A pose is a combination of position and orientation of the visual sensor such as camera which represents the translational and rotational matrix between a corresponding sequence of image locations.

In VSLAM system which is a major concern in this research, the location pose is computed by Visual Odometry (VO) which is somewhat similar to wheel odometry (Venkatachalapathy 2016; Nistér et al. 2004). VO is 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 endows them with time in- formation. TE estimates the path which a camera moves through over a certain period of time. There is a correlation between the VOTE and VSLAM, where VOTE focuses on constructing a consistent trajectory using a local map to obtain a highly accurate local trajectory, and VSLAM tackles global map consistency along the entire trajectory.

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 n points (Gao et al. 2003). There are many methods proposed for VOTE based on a PnP problem such as RANSAC. RANSAC 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 this method was introduced by Fischler and Bolles (Fis- chler & Bolles 1981) regarding the convergent speed and performance (Xing & Huang 2010; Zhang et al. 2011; Bhattacharya & Gavrilova 2013; Guo et al. 2015; Wang et al. 2016; Liu et al. 2017a).

However, the RANSAC performance is affected by the match ability of keypoints which are the main input for estimating the pose. The keypoints detection is a crucial part of VOTE because it provides the matching pair-keypoints and when the input space includes a lot of false matching keypoints, the estimation of the poses will be complicated and the RANSAC performance will be remarkably low. It is noted that the main source of the false matching keypoints "outliers" is the keypoints detection methods.

In a real environment, the captured scenes suffer from high variations, such as scaling, perspectives, illumination and scattered objects where the current approaches using a single visual descriptor cannot detect the prominent features to find their corresponding keypoints in other image frames.

The keypoints detection method needs to be robust and be able to find similar keypoints in the previous images regardless of any differences in image scaling, rotation, or variance of illumination. Keypoints 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 (Govender 2009; SHI et al. 2016).

Therefore, this chapter focuses on the keypoints detection stage in VOTE and proposes a new algorithm named MD-VOTE. The proposed method of a point selecting scheme provides the best points for feature matching. The three local feature descriptors, SURF, SIFT and ORB, are used to generate a potential combination of keypoints extracted from each image location with a proposed keypoint refinement method, whose function is to keep the distinctive keypoints and eliminate the overlapped keypoints, which will eventually reduce the trajectory estimation errors of a robot’s trip.

This technical report is organized as follows: Section 1.0 highlights the VOTE and its challenges in extracting the distinctive keypoints. Section 2.0 describes the proposed MD-VOTE algorithm. Section 3.0 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 4.0 gives a summary of the chapter.

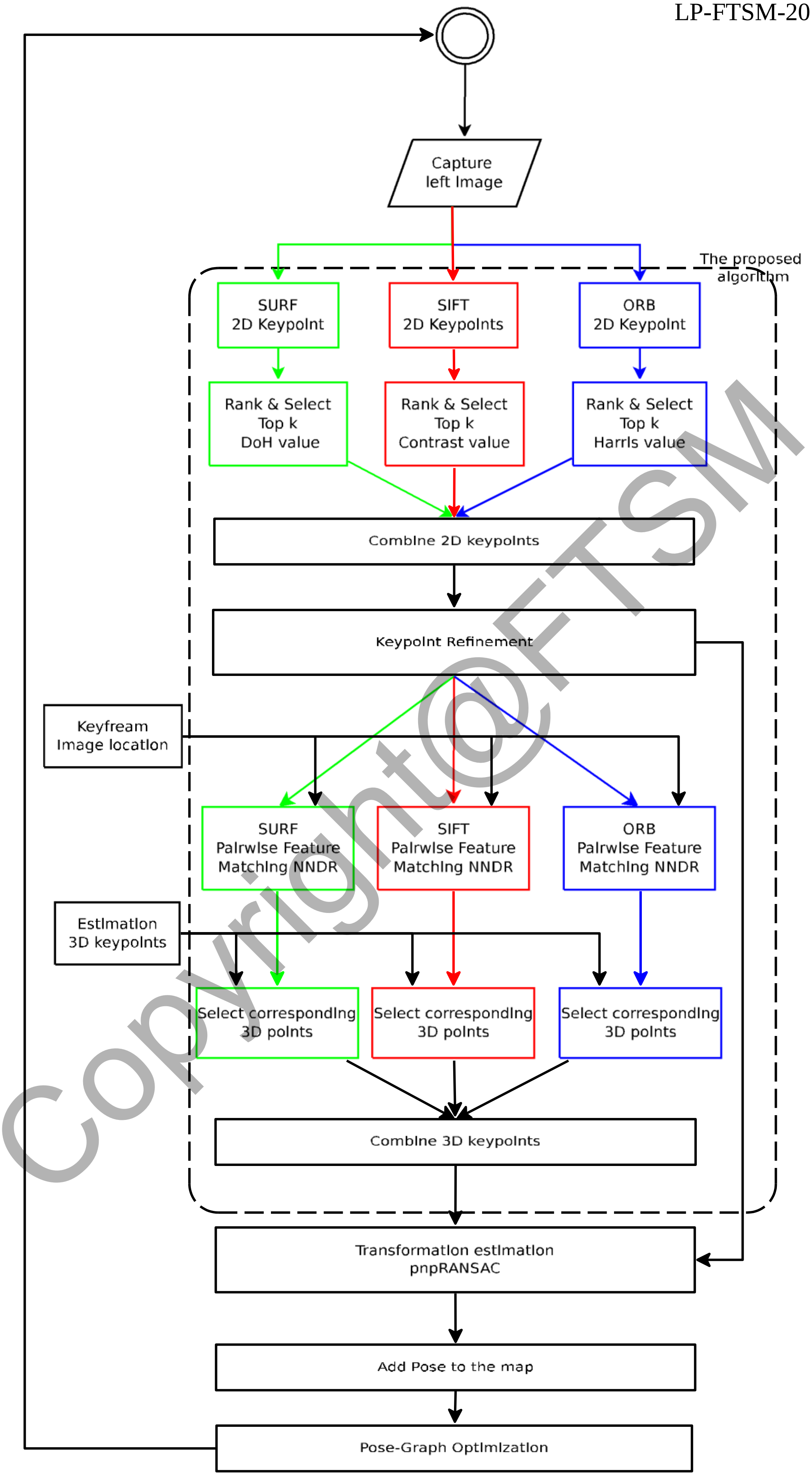
2.0 METHODOLOGY

In this section, a new VOTE algorithm named MD-VOTE is proposed to use the multi- ple 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 3D-2D matching keypoints which are extracted from the image locations by using the keypoint 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.

The following paragraph explains the proposed algorithm MD-VOTE relating to the traditional VOTE processes flowchart shown in Figure 2

2.1 Keypoints Detection

At first stage, the keypoint detector from each visual descriptor: SURF, SIFT and ORB detect their keypoints individually from the current image location. After that, the proposed algorithm ranks each keypoint according to its response value as described in paragraph a. followed by the refinement process as described in paragraph b.. Next, visual features are extracted from each refined keypoint, and the description of such keypoints are generated based on its corresponding descriptor. The final process in this stage is the key frame selection where the proposed algorithm selects the last frame, image location, available in the STM.



Capture left Image
The proposed algorithm
SURF 2D Keypoint
SIFT 2D Keypoints
ORB 2D Keypoint
Rank & Select Top k DoH value
Rank & Select Top k Contrast value
Rank & Select Top k Harris value
Combine 2D keypoints
Keypoint Refinement
Keyfream Image location
SURF Pairwise Feature Matching NNDR
SIFT Pairwise Feature Matching NNDR
ORB Pairwise Feature Matching NNDR
Estimation 3D keypoints
Select corresponding 3D points
Select corresponding 3D points
Select corresponding 3D points
Combine 3D keypoints
Transformation estimation pnpRANSAC
Add Pose to the map
Pose-Graph Optimization

Figure 1: The flowchart for the proposed MD-VOTE algorithm

Algorithm 1 : MD-VOTE

```
$kp_{SURF}$ is a set of 2D keypoints detected by SURF keypoint detector from the current image $I_t$
$kp_{SIFT}$ is a set of 2D keypoints detected by SIFT keypoint detector from the current image $I_t$
$kp_{ORB}$ is a set of 2D keypoints detected by ORB keypoint detector from the current image $I_t$
Ranked the keypoints $kp_{SURF}$ based on the Determinant of Hessian (DoH) value of each keypoint.
Ranked the keypoints $kp_{SIFT}$ based on the contrast value of each keypoint.
Ranked the keypoints $kp_{ORB}$ based on the Harris corner measure value of each keypoint.
$top_{SURF}$ is a set of the top k keypoints from $kp_{SURF}$
$top_{SIFT}$ is a set of the top k keypoints from $kp_{SIFT}$
$top_{ORB}$ is a set of the top k keypoints from $kp_{ORB}$
$kp_{all}$ is a set of the combination of keypoints $kp_{all} = top_{SURF} \cup top_{SIFT} \cup top_{ORB}$
$kp_{refined} = KeypointRefinement(kp_{all}, d_{min})$
$f = FeatureExtraction(kp_{refined}, SURF, SIFT, ORB)$, extracts the corresponding features of each keypoint based on its own descriptor.
$kp_{pair} = PairwiseMatching(kp_{refined}, f, I_{keyframe})$, where $I_{keyframe}$ is the keyframe image.
$kp_{3D}$ is a set of 3D keypoints extracted from a keyframe image based on the corresponding $kp_{pair}$.
$pose_t$ is the corresponding pose of the camera at time $t$ which is estimated by using $PnP - RANSAC(kp_{refined}, kp_{3D})$.
```

a. Ranking and Selecting Keypoints

The proposed algorithm starts with detecting three sets of keypoints using the three visual descriptors SURF, SIFT and ORB from the current image location. Each set of keypoints is ranked individually based on the keypoint response value as explained below:

1. The SIFT's keypoints detector applies the DoG in order to recognize as much as possible of the keypoints by generating several Gaussian-blurred images. These images are based on several scales of the input image.

After that, the SIFT's keypoints detector computes the DoG images based on the subtraction of neighbors in scale space from each other. Based on the DoG images, the keypoints are selected if they meet the following conditions: (1) they are locally extremal in the DoG images in space and scale. (2) They fulfil the threshold ratio of eigenvalues of the Hessian matrix. (3) The keypoints contrast is high. The keypoints which succeed are detected by interpolating through the DoG images (Lowe 2004). The contrast value is the keypoint response value which is used to determine how strong the keypoints are (Itseez 2014).

2. The SURF descriptor is partly inspired by SIFT, where the keypoints in SURF starts with computing integral 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 (Bay et al. 2008). The DoH value is the keypoint response value which is used to determine how strong the keypoints are(Itseez 2014).

3. The ORB develops oFAST for the keypoints detector which enhances the FAST detector. The oFAST detects the keypoints 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 keypoints are sorted out based on Harris corner computations to select the top keypoints (Rublee et al. 2011b). The Harris corner computation produces the keypoint response value which is used to determine how strong the keypoints are (Itseez 2014).

After the ranking of each keypoint set, the top k keypoints are selected from each keypoint set and are combined using the following equation:

$$kp_{all} = top_{SURF} \cup top_{SIFT} \cup top_{ORB} \qquad (1)$$

where top_{SURF}, top_{SIFT}, top_{ORB} are the sets of the top k keypoints extracted from each descriptor. Finally, the set kp_{all} is passed to the next process to keep the distinctive keypoints and eliminate the overlapped keypoints as described in the next paragraph.

b. Keypoint Refinement

The result of the previous process is the set of keypoints kp_{all} which contains the top k keypoints detected by each descriptor. In this stage, the keypoint refinement method keeps the distinctive keypoints 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 dmin.

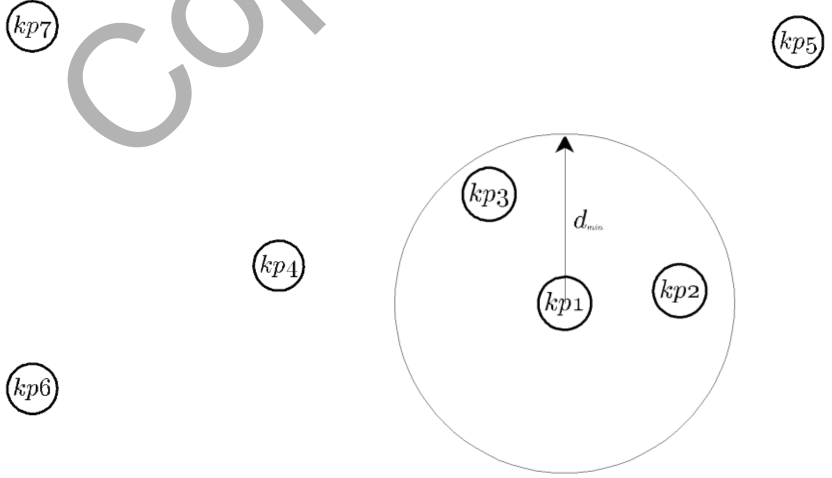


Figure 2: A simple example for the refining method.

Figure 2.0 shows an example of the refining method as follows:

1. Sorts out the keypoints in the set kp_{all} according to the keypoint location in the image.
2. kp_d is a set of keypoints which includes the first keypoint kp_0 in the set kp_{all}.
3. The set kp_d also includes all keypoints lying within a distance d_{min} from the keypoint kp_0.
4. The function selects the keypoint from the set kp_d which has the highest rank based on its corresponding descriptor. As shown in Figure 2.0, the set $k\,p_d$ in- cludes the keypoints $[k\,p_1, k\,p_2, k\,p_3]$ whose corresponding ranks are $[2, 9, 5]$ re- spectively. The keypoint $k\,p_2$ is selected because it obtained the highest rank as the keypoint with the highest rank is more distinctive and traceable.
5. The selected keypoint from the previous step is added to the set kp_{refine} and the set kp_d is eliminated from the set kp_{all}.
6. The steps (2,3,4,5) are repeated until kp_{all} is empty

2.2 Keypoints Tracking

The second stage handles the pairwise matching of the keypoints and checks the mutual consistency. After refining the combined keypoints, each descriptor processes the pairwise matching of its keypoints. This set of keypoints is pairwise matched between the current image location and the key frame image location using the NNDR approach with k-d tree and produces the set of matched pair-keypoints (Labbe & Michaud 2013). Then, the proposed algorithm extracts the 3D points from the key frame based on the corresponding 2D keypoints. At this stage, the refined 2D keypoints and there corresponding 3D keypoints are passed on to the next stage of motion estimation.

2.3 Motion Estimation

The motion estimation stage is the stage where the pose is estimated between the cur- rent image location and the key frame location. The refined 2D keypoints and their corresponding 3D keypoints 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.

3.0 Experimental Results and Discussion

The proposed MD-VOTE algorithm performance was evaluated using three outdoor scenes: the sequences 00, 02 and 05 from the public dataset KITTI (Geiger et al. 2012b). 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 (Fanfani et al. 2016; Mur-Artal & Tardós 2017; Mur- Artal et al. 2015; Pire et al. 2015; Engel et al. 2014).

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. The evaluation criteria calculate the translational error using the segments of the trajectory at 100, 200,. . . , 800m lengths.

4.0 Experimental Setup

The performance evaluation of the proposed algorithm MD-VOTE on KITTI outdoor environments uses the KITTI Vision Benchmark Suite (Geiger et al. 2012b). This experiment applies to the following sequences: (1) Seq-00 with 4541 images over the distance of 3714m, (2) Seq-02 with 4661 images over the distance of 5075m and (3) Seq-05 with 2761 images over the distance of 2223m. These sequences are the longest path in the KITTI dataset which can show the efficiency and robustness of the proposed algorithm in long paths.

All the parameters of the visual descriptors SURF. SIFT and ORB are set as reported in OpenCV (Itseez 2014). The value of the variable maximum number of keypoints extracted from the image being determined at 400 points for each descriptor. This value is a default set by RTAB-Map and used as the previously proposed algorithms EBF-AL and EBF-APL. For comparative purposes, RTAB-Map is used as a VOTE benchmark, where its parameters are set up as reported in (Labbe & Michaud 2014; Labbé 2014).

4.1 Trajectory Estimation Results and Discussion

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 keypoints: 400 keypoints and 1000 keypoints. This experiment is conducted on the same three sequences (00, 02 and 05) in the KITTI dataset. Table 1 shows the results of the first experiment conducted using 400 and 1000 keypoints 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 keypoints used in trajectory estimation. In contrast, with a maximum of 400 extracted keypoints, 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 is capable of extracting and matching the 400 keypoints more efficiently than other feature descriptors which improve the PnP-RANSAC process in estimating the poses.

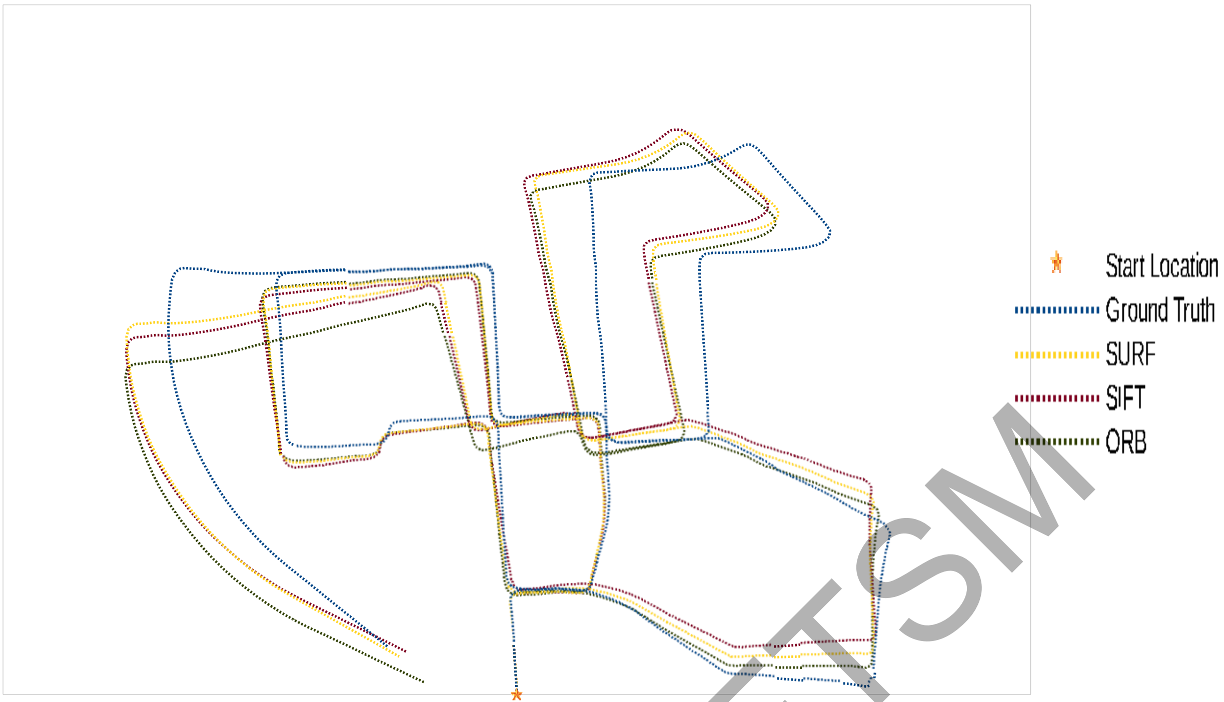
Significantly, the SURF with 400 keypoints has scored the best results in the trajectory estimation for the sequences 00 and 02 over the other descriptors even with a different number of extracted keypoints. This fact shows that the quality of keypoints is more significant than the quantity in estimating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

Figures 3, 4 and 5 show the trajectory estimation for the sequences:00, 02 and 05 respectively using the three visual descriptors SURF, SIFT and ORB in addition to the ground truth trajectory. The figures show that the estimated trajectories using 1000 keypoints are nearly the same even though they are less accurate than the estimated trajectories using 400 keypoints. There is a remarkable disparity between the estimated trajectories using 400 keypoints over the estimated trajectories using 1000 keypoints.

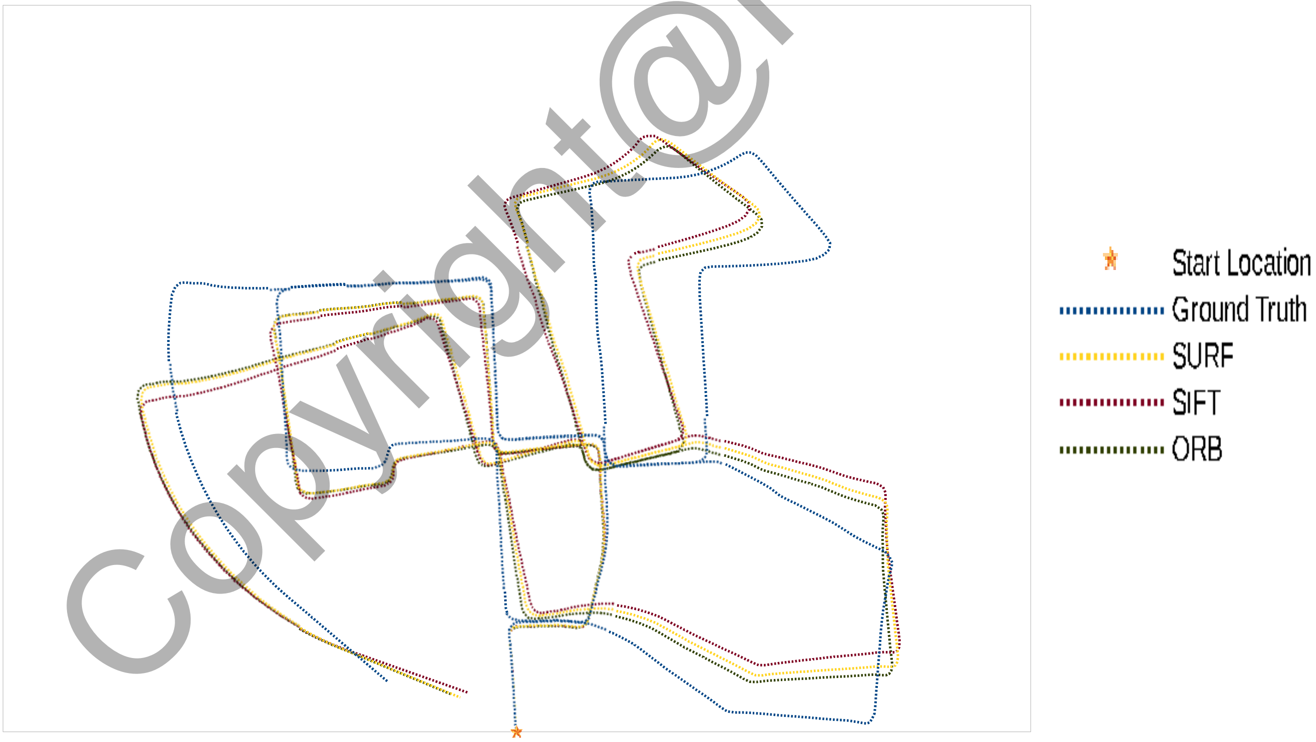
Table 1: Trajectory estimation errors for sequences 00, 02 and 05 using the benchmark VOTE "RTAB-Map" with different visual descriptors with the extracted 400 and 1000

keypoints. The average translational error is shown in percentages and the average rotational error is shown in degrees per meter.

No. Keypoints	KITTI Sequence	Seq-00, 4541 Images		Seq-02, 4661 Images		Seq-05, 2761 Images	
	Visual descriptor	Translation %	Rotation d/m	Translation %	Rotation d/m	Translation %	Rotation d/m
400	SURF	0.016375	0.000113	0.013978	7.4E-05	0.010763	6.5E-05
	SIFT	0.019614	0.000141	0.017221	9.9E-05	0.01249	6.8E-05
	ORB	0.017698	0.000116	0.015105	9.6E-05	0.011001	6.4E-05
1000	SURF	0.025192	0.000138	0.015103	8.38E-05	0.009958	0.000056
	SIFT	0.025316	0.000152	0.01533	9.93E-05	0.01165	0.00007
	ORB	0.02509	0.000137	0.015001	0.000082	0.009754	0.000054



(a)



(b)

Figure 3: Trajectory estimation for KITTI Sequence-00, 4541 images using different feature descriptors: (a) using 400 keypoints. (b) using 1000 keypoints

The second experiment is conducted to evaluate the performance of the proposed MD-VOTE algorithm with the value of a different radius d_{min}, and shows the impact of a

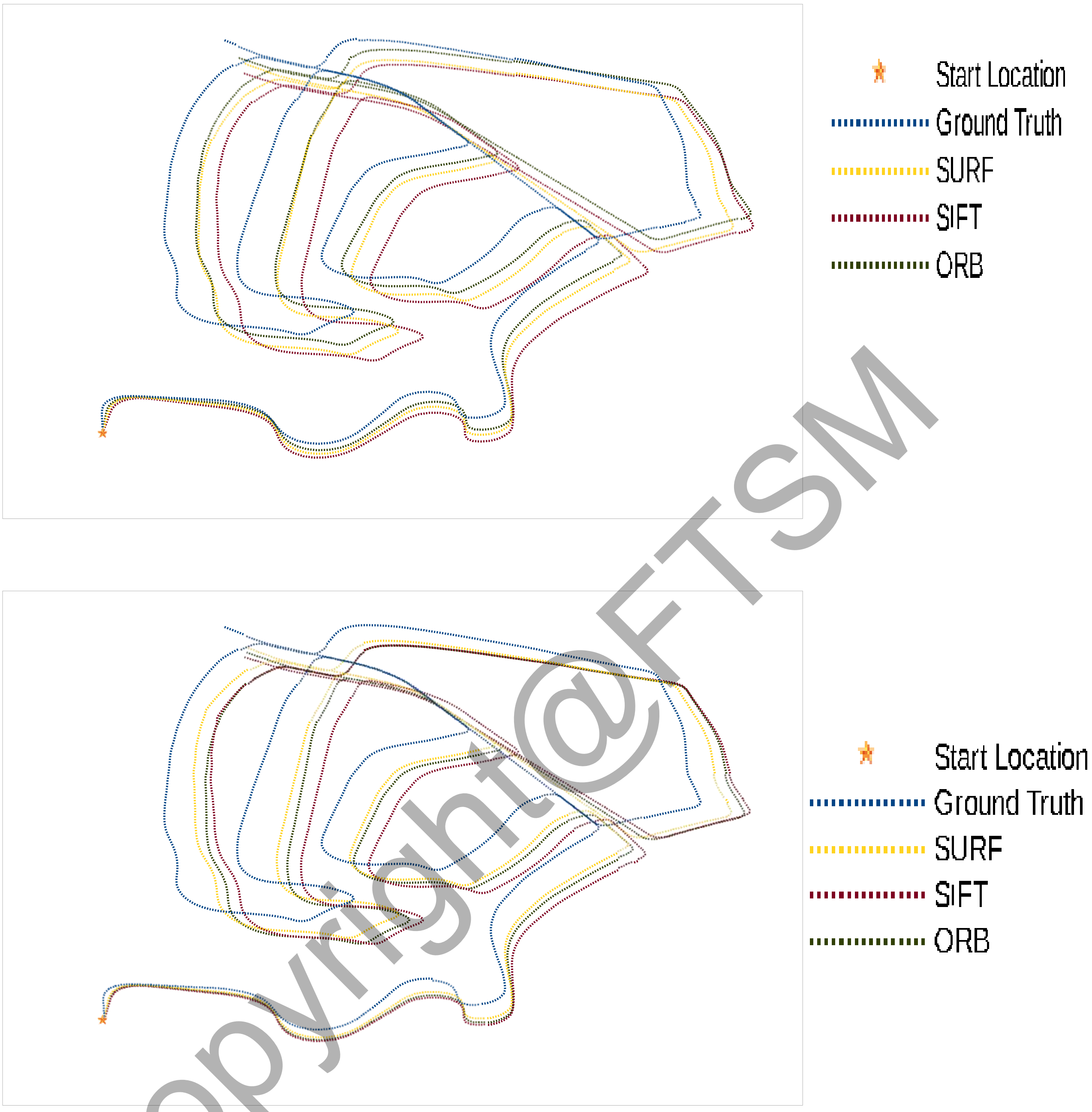
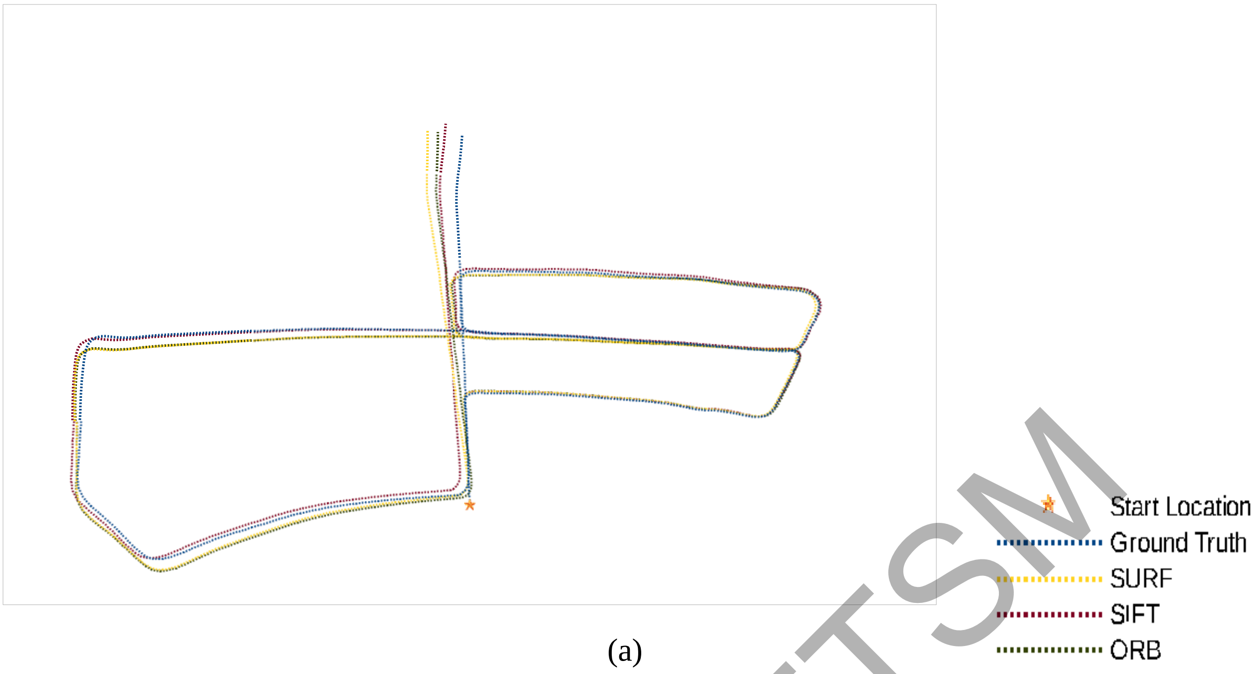
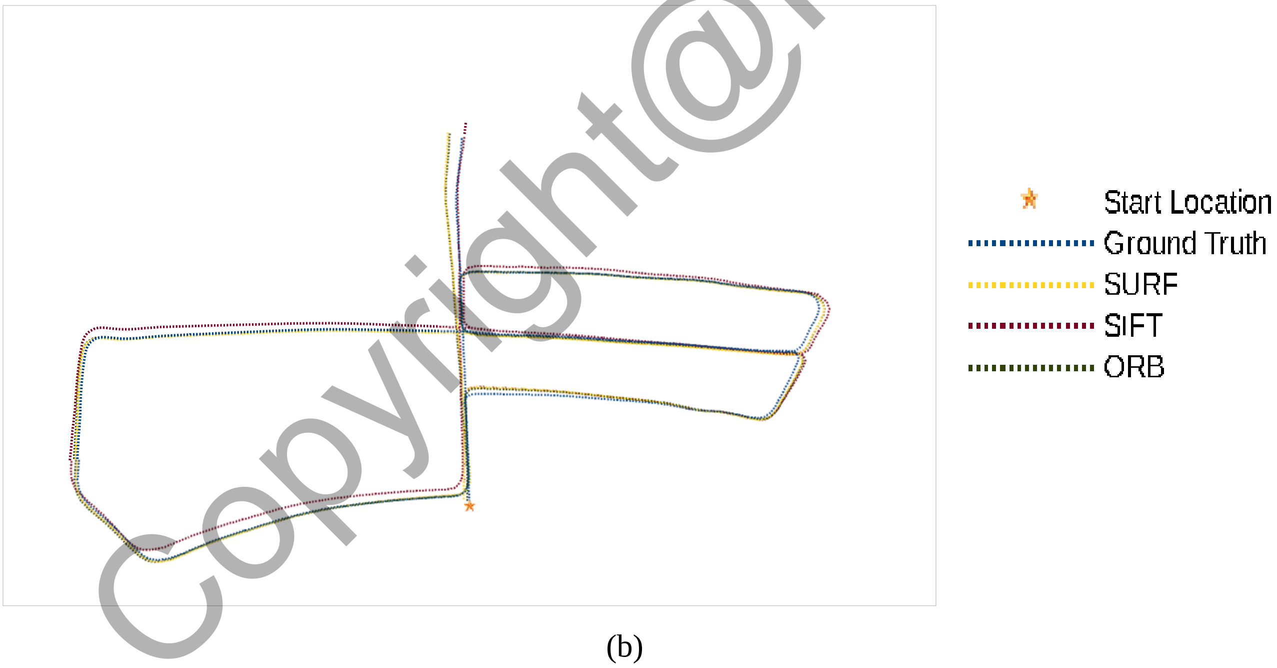


Figure 4: Trajectory estimation for KITTI Sequence-02, 4661 images using different feature descriptors: (a) using 400 keypoints. (b) using 1000 keypoints



(a)



(b)

Figure 5: Trajectory estimation for KITTI Sequence-05, 2761 images using different feature descriptors: (a) using 400 keypoints. (b) using 1000 keypoints

Table 2: Trajectory estimation errors for sequence 05 using MD-VOTE with different radius values. No. Keypoints = Maximum number of keypoints detected from the images different radius on the efficiency of the proposed algorithm in estimating the trajectory.

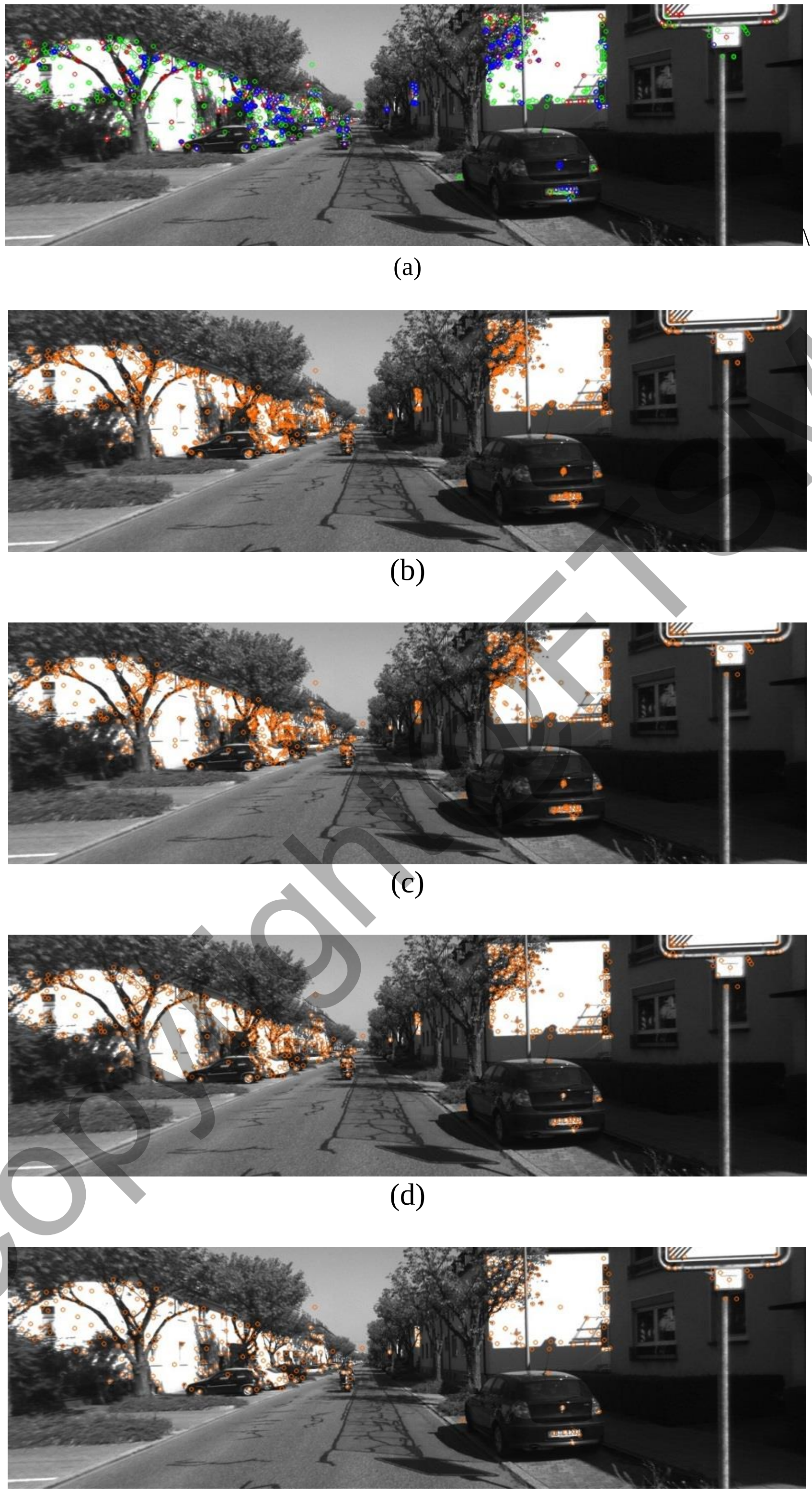
Distance in pixel	No. Keypoints	Translation %	Rotation d/m
No filter	1200	0.010084	6.1E-05
$d_{min} = 1$	980	0.008955	5.4E-05
$d_{min} = 3$	719	0.010589	5.9E-05
$d_{min} = 6$	518	0.011374	6.6E-05
$d_{min} = 9$	266	0.012149	8.2E-05

Table 2 shows the trajectory estimation errors of the proposed MD-VOTE algorithm using sequence 05 with 2761 images from KITTI datasets. Figure 7 shows the trajectories which are estimated by the proposed MD-VOTE using different radius values against the ground truth.

Figure 6 shows an example of the number of keypoints detected by the three visual descriptors SURF, SIFT and ORB, and shows the impact of keypoint refinement method using a different radius on the number of keypoints. The image location in the example is the location number 5 taken from sequence 05 of the KITTI dataset.

It is noticed that using the proposed MD-VOTE algorithm without refining the keypoints which are 1200 keypoints 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 keypoints refinement using radius $d_{min} = 1$, the maximum number of keypoints reaches 980 keypoints 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 keypoints decreases and the errors rate increases.

Figures 8.a and 8.b show the translational and rotational errors as a function of the path length and the trajectory segmented at 100, 200,. . . , 800m lengths (Geiger et al. 2012a)



(a)

(b)

(c)

(d)

(e)

Figure 6: Example for the keypoint refinementmethod using a different radius

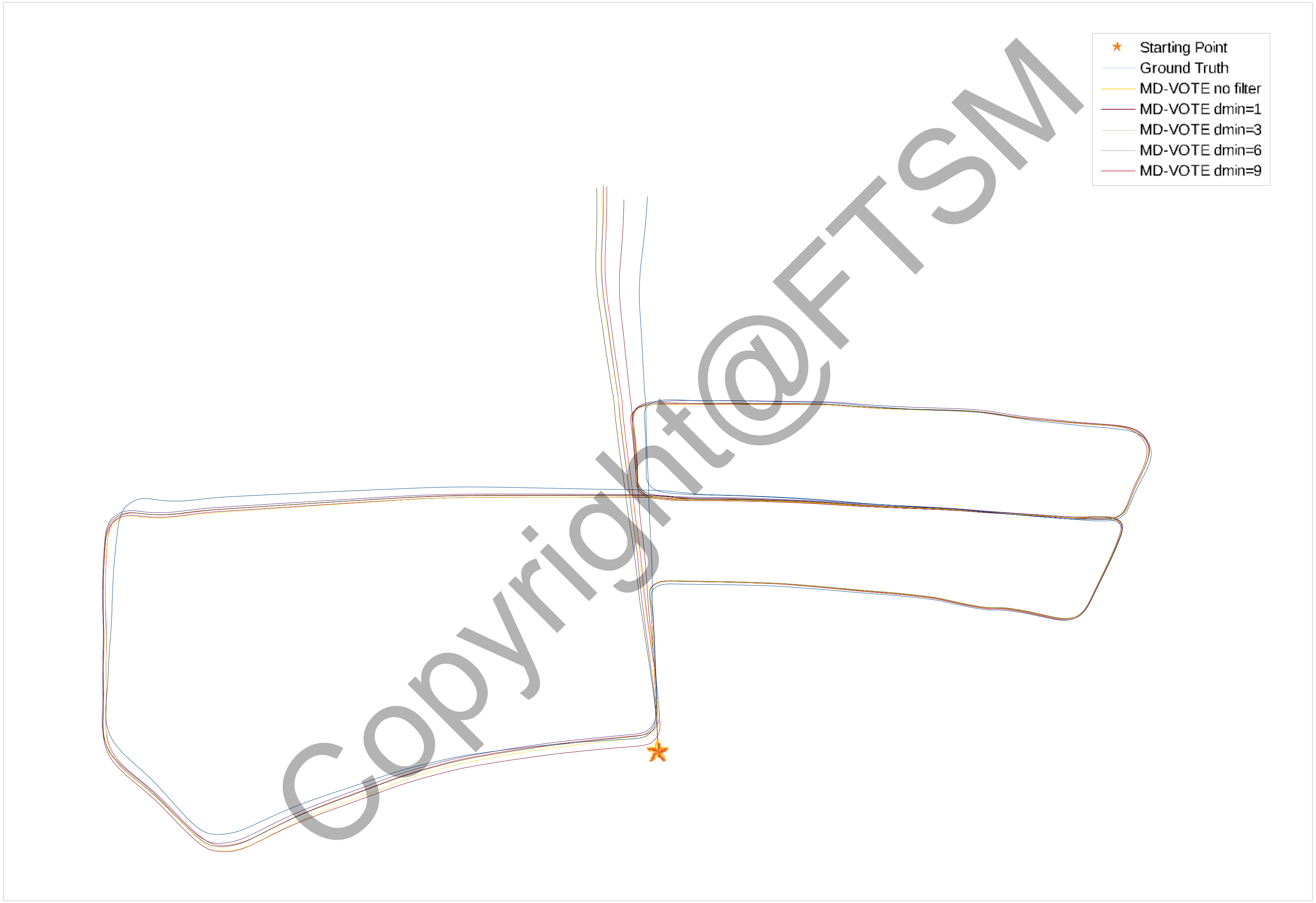
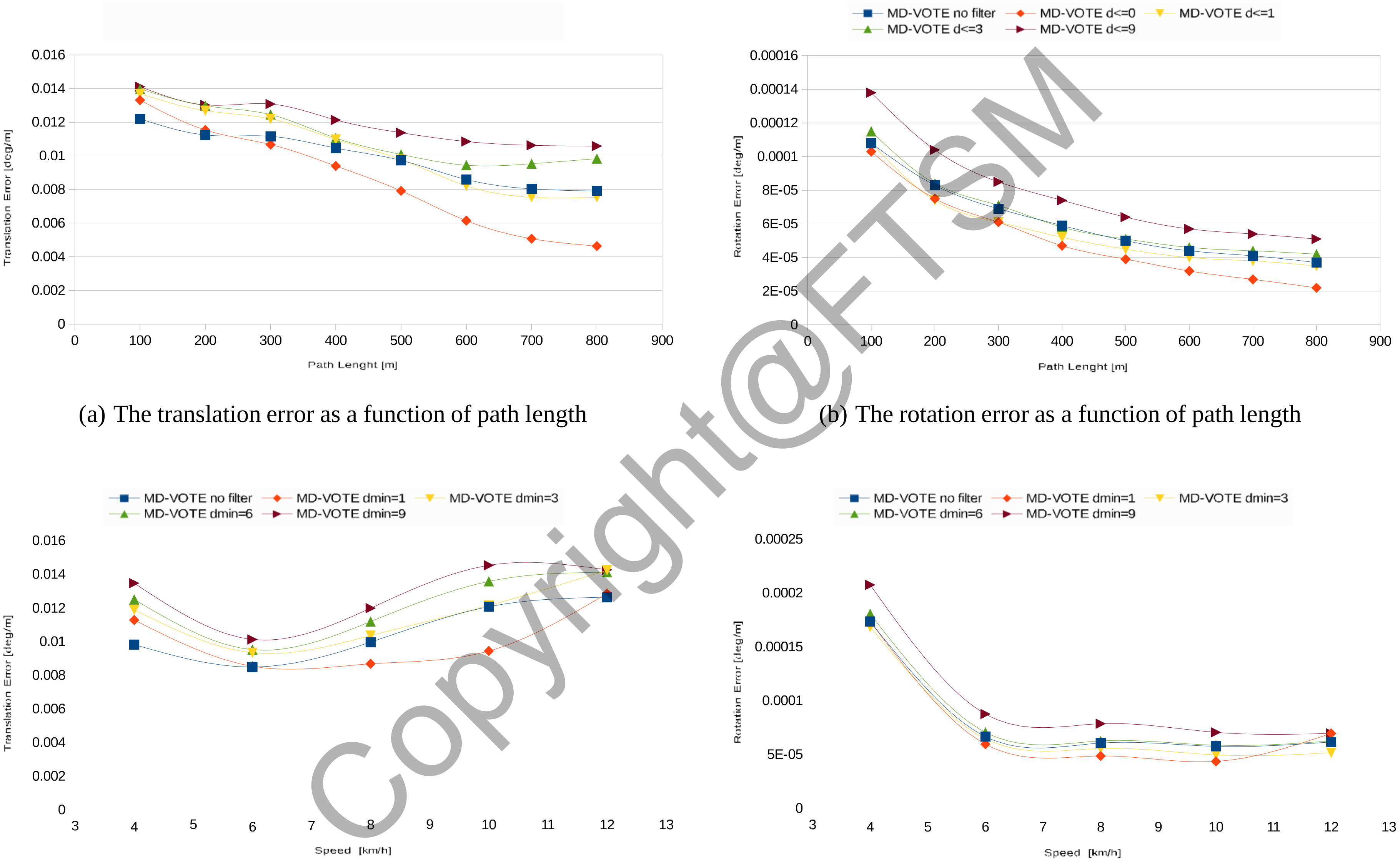


Figure 7: Trajectory estimation for sequence 05 using MD-VOTE with different radius values.



(a) The translation error as a function of path length

(b) The rotation error as a function of path length

(c) The translation error as a function of speed

(d) The rotation error as a function of speed

Figure 8: The translational and rotational errors as a function of speed and path length for sequence 05 using MD-VOTE with different radius values

It is noticed that the proposed MD-VOTE algorithm used with the keypoint 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, Figures 8.c and 8.d 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 Figure 8.c 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, Figure 8.d 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 2, 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 keypoint refinement method using radius $d_{min} = 1$, unless it is explicitly stated otherwise.

The final experiment evaluates the performance of the proposed MD-VOTE algorithm and compares between the proposed MD-VOTE with, and without the keypoint refinement method against the standard RTAB-Map in trajectory estimation for the three sequences 00, 02 and 05 from the KITTI dataset. Table 3 shows the average translational and average rotational errors for each sequence. The last row in Table 3 shows the relative change ratio calculated between the proposed MD-VOTE algorithm with the keypoint refinement method and the standard RTAB-Map. For further information about relative change, see Section d..

To clarify the efficiency of the proposed MD-VOTE algorithm, a comparison has been made between MD-VOTE and RTAB-Map in terms of relative change with

Table 3: Trajectory estimation errors for sequences 00, 02 and 05 using the proposed MD-VOTE with and without the filtering method against RTAB-Map

Algorithm name	No. of Keypoints	Seq-00, 4541 Images		Seq-02, 4661 Images		Seq-05, 2761 Images	
		Translation %	Rotation d/m	Translation %	Rotation d/m	Translation %	Rotation d/m
RTAB-Map	1000	0.024504%	0.000126	0.014704%	0.000077	0.010315%	0.000063
MD-VOTE without filter	1200	0.014994%	0.000101	0.017034%	0.000099	0.010113%	0.000064
MD-VOTE with filter	980	0.013636%	0.000096	0.013432%	0.000078	0.008955%	0.000054
Relative change MD-VOTE vs RTAB-Map	-2%	-44.35%	-23.80%	-8.65%	+1.29%	-13.18%	-14.28%

respect to trajectory estimation errors. The results of the comparison based on sequence 00, Table 3 shows 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.

Figure 9 shows the trajectory estimation for the sequence 00 constructed by the three algorithms MD-VOTE with, and without filtering against RTAB-Map whereas Figures 10.a and 10.b show the translational and rotational errors as function of path length, and Figures 10.c and 10.d show the translational and rotational errors as function of moving speed. The proposed MD-VOTE algorithm has scored the least errors in both translational and rotational errors (0.013636%, 9.6E-05 d/m) respectively. We wish to compare these trajectory errors with respect to the trajectory errors estimated by RTAB-Map,

Figure 11 shows the trajectory estimation for the sequence 02 constructed by the three algorithms MD-VOTE with, and without filtering against RTAB-Map whereas Figures 12.a and 12.b show the translational and rotational errors as function of path length and Figures 12.c and 12.d show the translational and rotational errors as function of moving speed. The proposed MD-VOTE algorithm has scored the least errors in both translational and rotational errors (0.013432%, 7.8E-05 d/m) respectively.

Third, Figure 13 shows the trajectory estimation for the sequence 05 constructed by the three algorithms (MD-VOTE with, and without filtering against RTAB- Map) whereas Figures 14.a and 14.b show the translational and rotational errors as function of path length and Figures 14.c and 14.d show the translational and rotational errors as function of moving speed. The proposed MD-VOTE algorithm has scored the least errors in both translational and rotational errors (0.008955%, 5.4E-05 d/m) respectively.

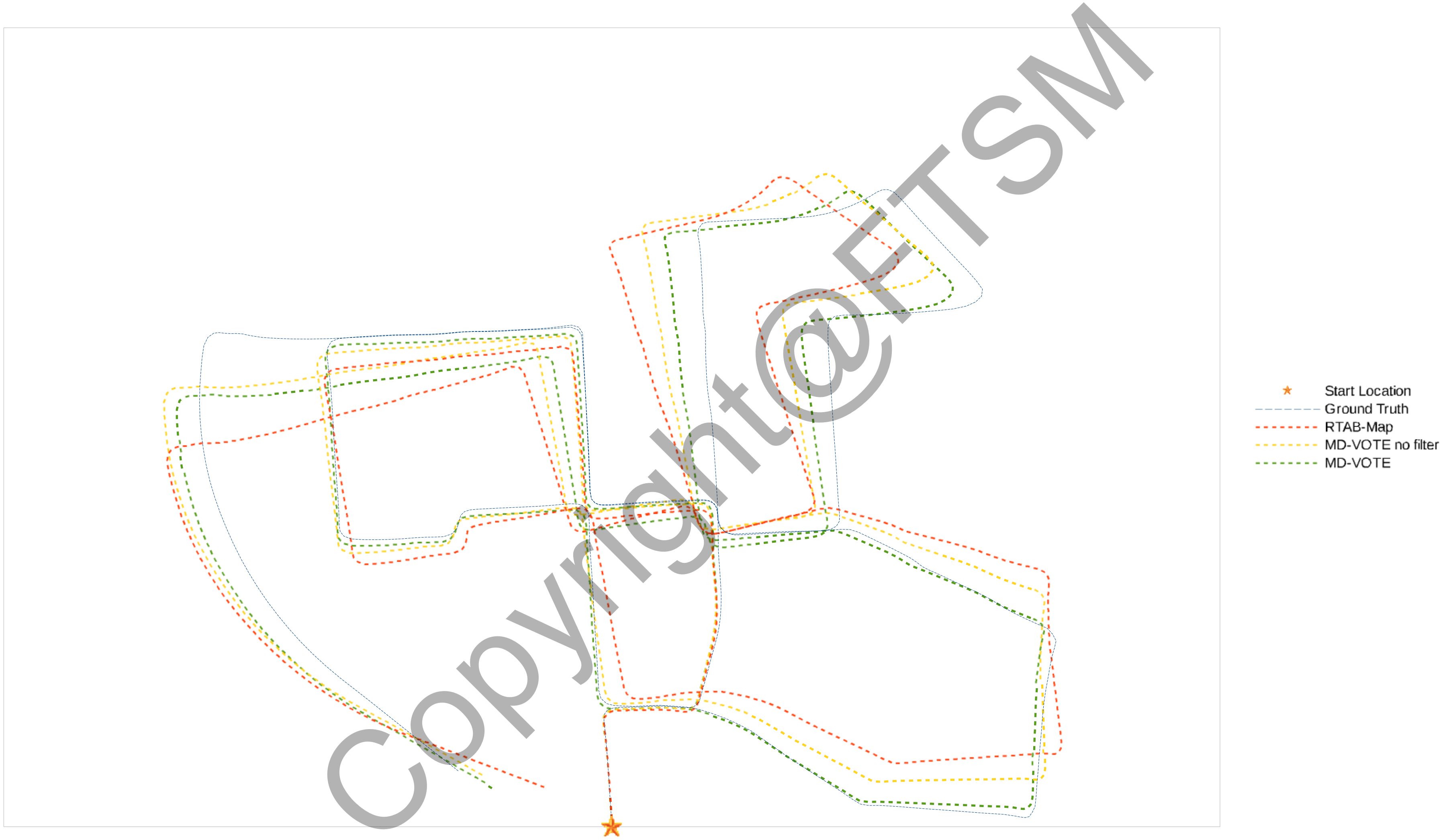
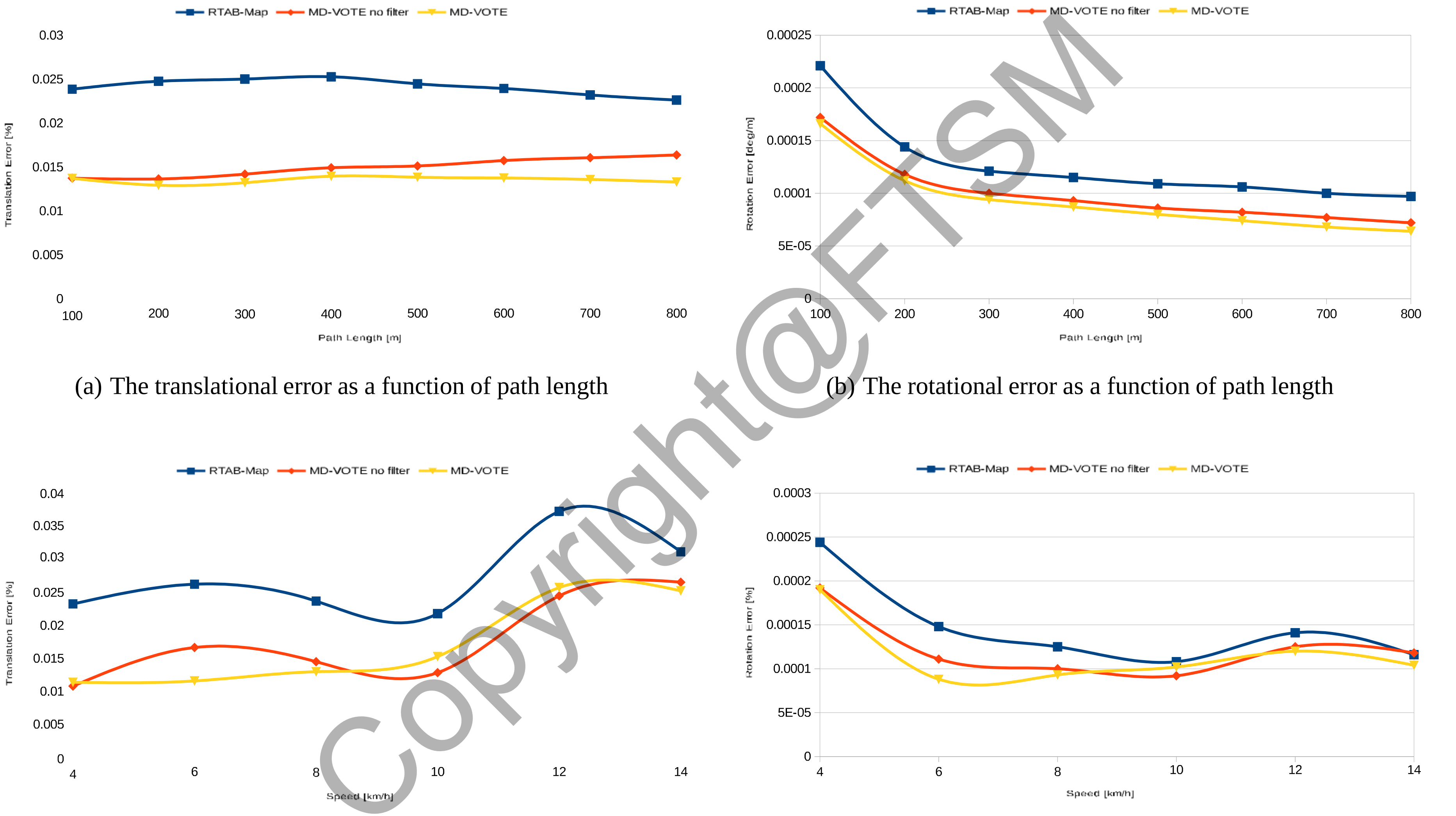


Figure 9: Trajectory estimation errors for sequences 00 using the proposed MD-VOTE with and without the filtering method against RTAB-Map



(a) The translational error as a function of path length

(b) The rotational error as a function of path length

(c) The translational error as a function of speed

(d) The rotational error as a function of speed

Figure 10: The translational and rotational errors as a function of speed and path length for sequence 00 using the proposed MD-VOTE with and without the filtering method against RTAB-Map.

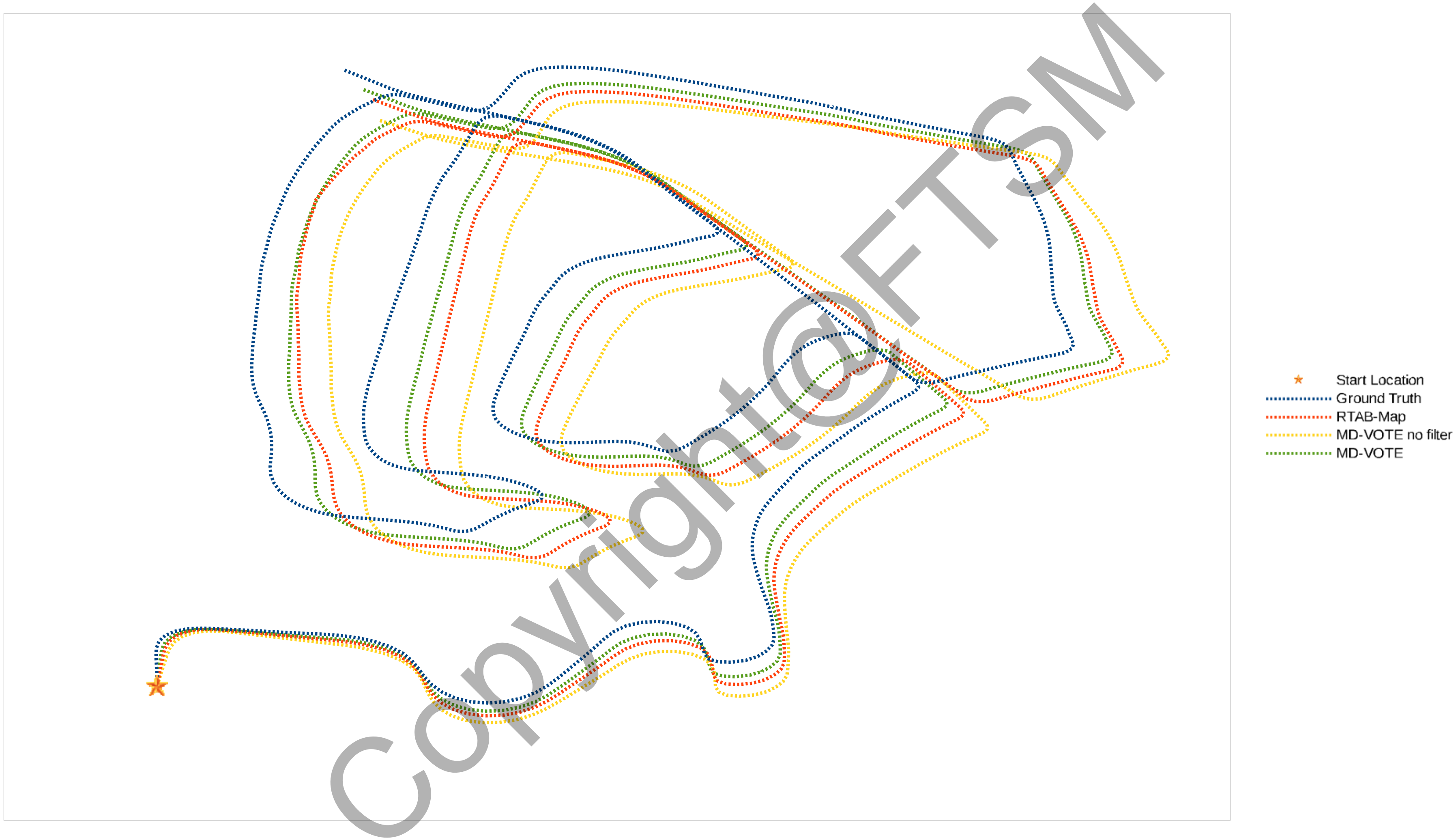
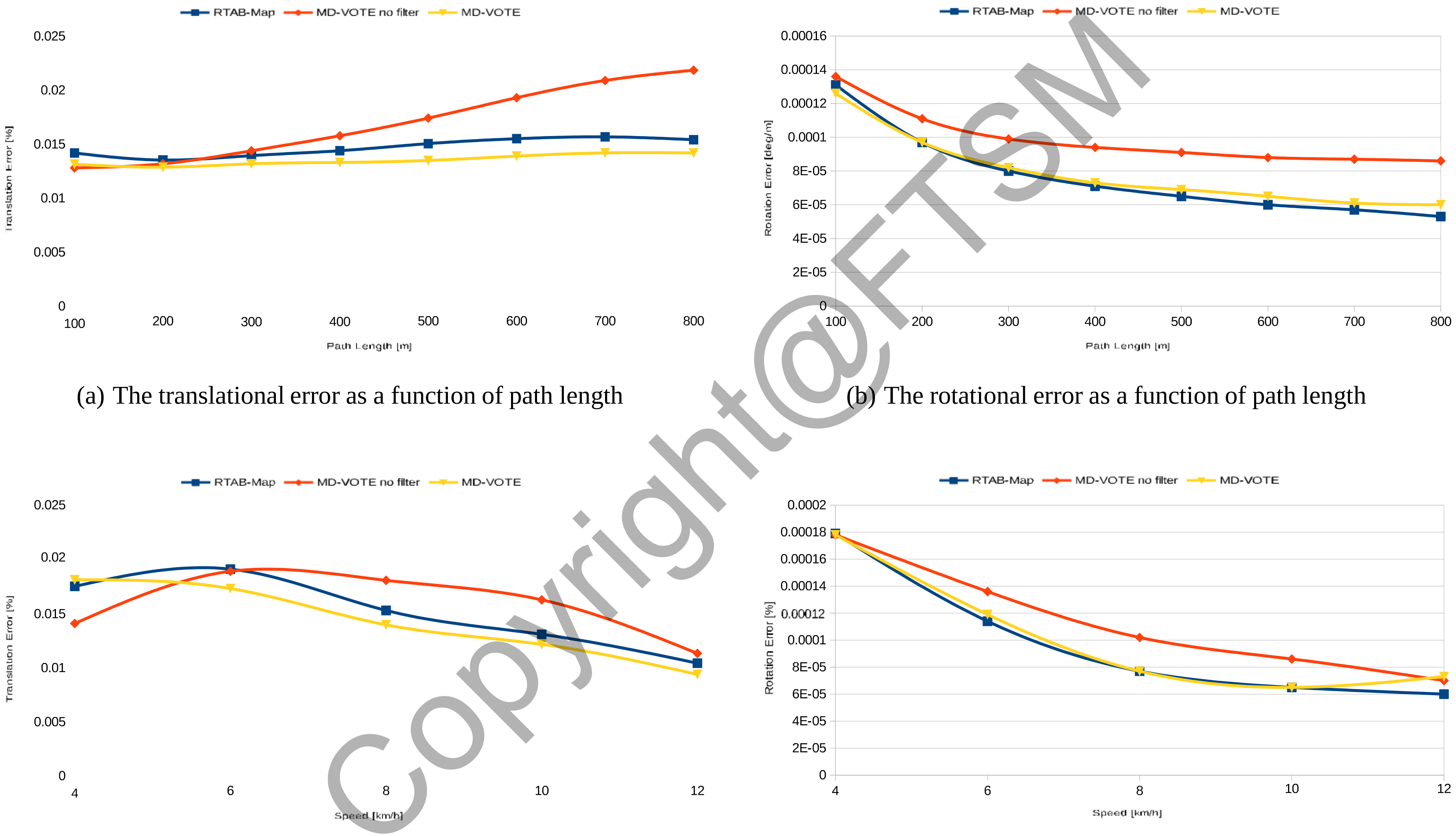


Figure 11: Trajectory estimation errors for sequences 02 using the proposed MD-VOTE with and without the filtering method against RTAB-Map



(a) The translational error as a function of path length

(b) The rotational error as a function of path length

(c) The translational error as a function of speed

(d) The rotational error as a function of speed

Figure 12: The translational and rotational errors as a function of speed and path length for sequence 02 using the proposed MD-VOTE with and without the filtering method against RTAB-Map.

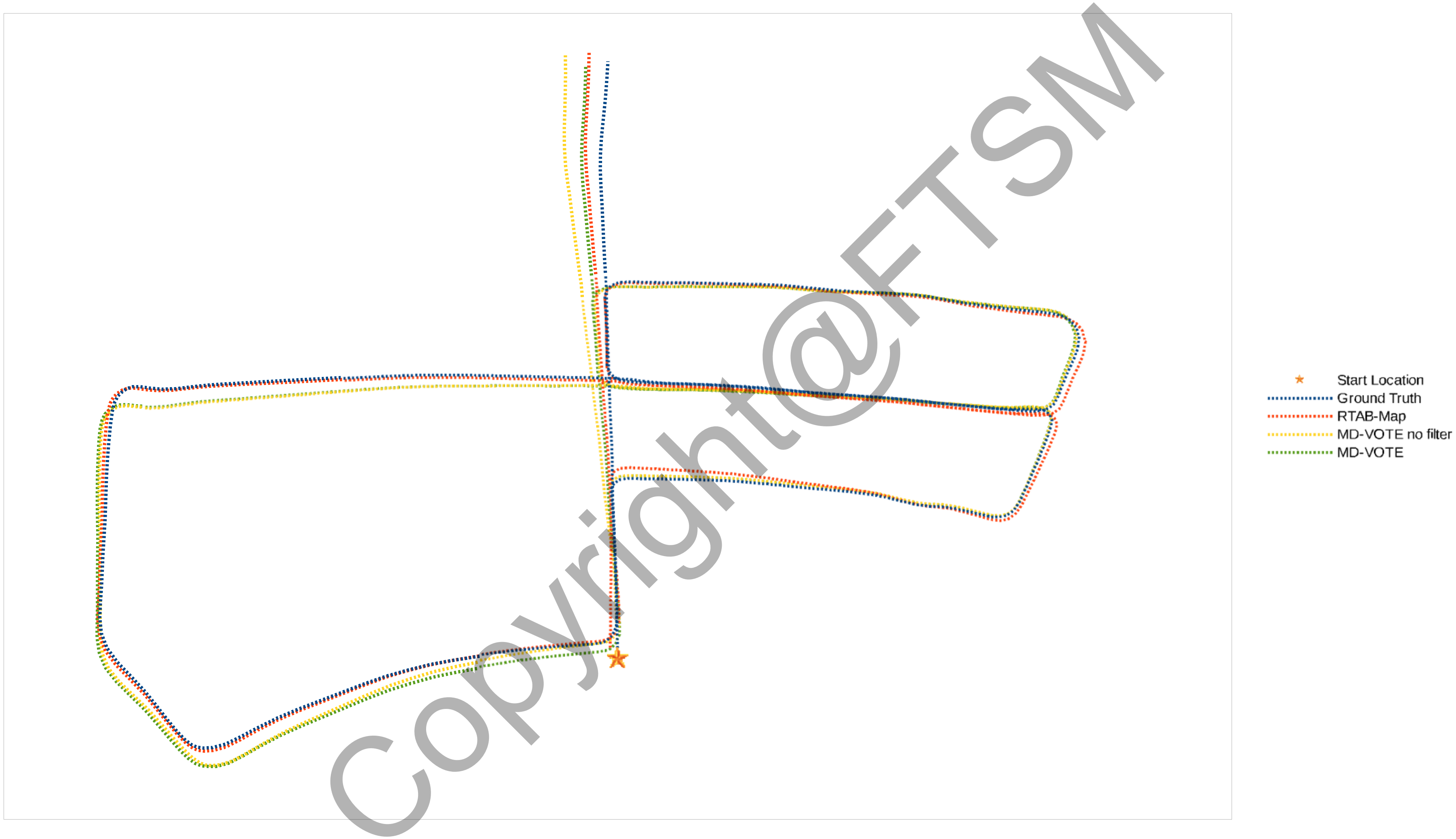
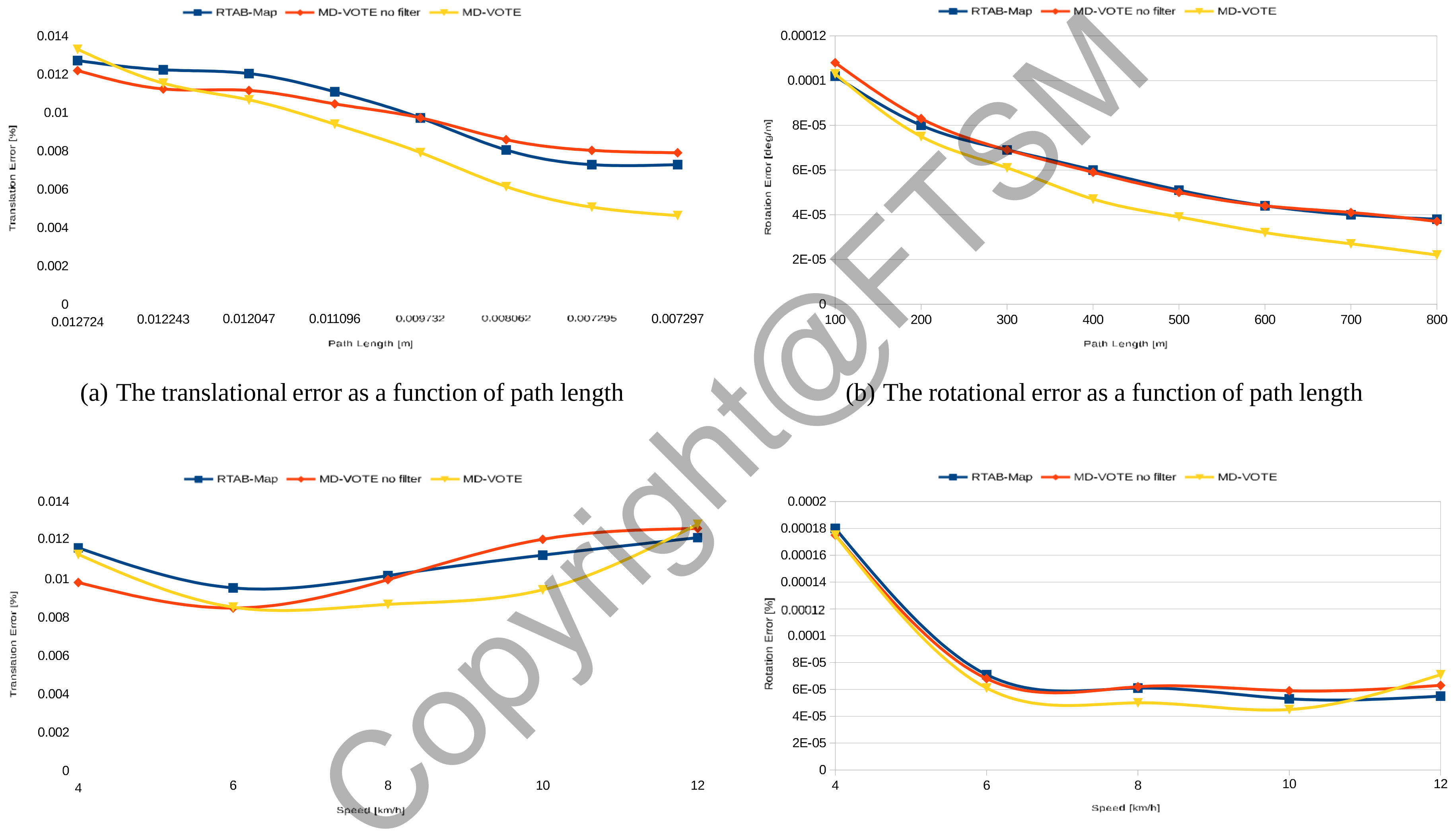


Figure 13: Trajectory estimation errors for sequences 05 using the proposed MD-VOTE with and without the filtering method against RTAB-Map



(a) The translational error as a function of path length

(b) The rotational error as a function of path length

(c) The translational error as a function of speed

(d) The rotational error as a function of speed

Figure 14: The translational and rotational errors as a function of speed and path length for sequence 05 using the proposed MD-VOTE with and without the filtering method against RTAB-Map.

4.0 SUMMARY

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

ACKNOWLEDGEMENT

The authors would like to extend their appreciation and gratitude to FRGS/1/2016/ICT02/UKM/02/5 and GP-K007762 grants for funding this project.

REFERENCES

Bay, H., Ess, A., Tuytelaars, T. & Van Gool, L. 2008. Speeded-up robust features (surf) Computer vision and image understanding 110(3): 346–359.

Bhattacharya, P. & Gavrilova, M. 2013. Dt-ransac: a delaunay triangulation based scheme for improved ransac feature matching. Transactions on Computational Science XX, pp. 5–21. Springer.

Engel, J., Schöps, T. & Cremers, D. 2014. Lsd-slam: Large-scale direct monocular slam. Computer Vision–ECCV 2014, pp. 834–849. Springer.

Fanfani, M., Bellavia, F. & Colombo, C. 2016. Accurate keyframe selection and key- point tracking for robust visual odometry. Machine Vision and Applications .

Fischler, M.A. & Bolles, R.C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commu- nications of the ACM 24(6): 381–395.

Gao, X.S., Hou, X.R., Tang, J. & Cheng, H.F. 2003. Complete solution classification for the perspective-three-point problem. IEEE transactions on pattern analysis and machine intelligence 25(8): 930–943.

Geiger, A., Lenz, P. & Urtasun, R. 2012b. Are we ready for autonomous driving? the kitti vision benchmark suite. Conference on Computer Vision and Pattern Recognition (CVPR).

Geiger, A., Lenz, P. & Urtasun, R. 2012a. Are we ready for autonomous driving? the kitti vision benchmark suite. Computer Vision and Pattern Recognition (CVPR) 2012 IEEE Conference on, pp. 3354–3361. IEEE.

Govender, N. 2009. Evaluation of feature detection algorithms for structure from motion. 3rd Robotics and Mechatronics Symposium (ROBMECH 2009). Pretoria, South Africa .

Guo, J., Wei, Z. & Miao, D. 2015. Lane detection method based on improved ransac algorithm. Autonomous Decentralized Systems (ISADS), 2015 IEEE Twelfth International Symposium on, pp. 285–288. IEEE.

Itseez. 2014. The OpenCV Reference Manual. Itseez, 2.4.9.0 ed.

Labbe, M. & Michaud, F. 2013. Appearance-based loop closure detection for online large scale and long-term operation. Robotics, IEEE Transactions on 29(3): 734–745.

Liu, L., Wang, Y., Zhao, L. & Huang, S. 2017b. Evaluation of different slam algorithms using google tangle data. IEEE Conference on Industrial Electronics and Applications.

Lowe, D.G. 2004. Distinctive image features from scale-invariant keypoints. International journal of computer vision 60(2): 91–110.

Mur-Artal, R. & Tardós, J.D. 2017. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. IEEE Transactions on Robotics 33(5) 1255–1262. doi:10.1109/TRO.2017.2705103.

Nistér, D., Naroditsky, O. & Bergen, J. 2004. Visual odometry. Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Com- puter Society Conference on, vol. 1, pp. I–I. Ieee.

Pire, T., Fischer, T., Civera, J., De Cristóforis, P. & Berlles, J.J. 2015. Stereo parallel tracking and mapping for robot localization. Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pp. 1373–1378. IEEE.

Rublee, E., Rabaud, V., Konolige, K. & Bradski, G. 2011b. Orb: an efficient alternative to sift or surf. Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 2564–2571. IEEE.

SHI, D.c., DONG, X.c. & ZHENG, Y. 2016. An improved orthogonal iterative algorithm for monocular camera pose estimation. DEStech Transactions on Com- puter Science and Engineering 3(aics).

Venkatachalapathy, V. 2016. Visual Odometry Estimation Using Selective Features. Rochester Institute of Technology.

Wang, Y., Zheng, J., Xu, Q.Z., Li, B. & Hu, H.M. 2016. An improved ransac based on the scale variation homogeneity. Journal of Visual Communication and Image Representation 40: 751–764.

Xing, C. & Huang, J. 2010. An improved mosaic method based on sift algorithm for uav sequence images. Computer Design and Applications (ICCDA), 2010 Inter- national Conference on, vol. 1, pp. V1–414. IEEE