Landmarks Based Human-like Guidance for Driving Navigation in an Urban Environment

Bihao Wang, Quentin Stafford-Fraser, Peter Robinson
Computer Laboratory, University of Cambridge, UK.
Email: {firstname.lastname}@cl.cam.ac.uk

Eduardo Dias, Lee Skrypchuk
Jaguar Land Rover Coventry, UK.
Email: {edias, lskrypch}@jaguarlandrover.com

***Abstract*—Driving is a cognitively demanding task, and many current navigation systems present confusing guidance instructions that add to the distraction. Human navigators, by contrast, schedule their advice to minimise distraction, and phrase instructions in terms of visible landmarks to avoid confusion. In this paper, we present the basis for a 'natural navigation' system which interprets distances as references to landmarks. We use Extended Kalman Filtering to integrate visual odometry with other sensor data in order to obtain precise vehicle motion, then, based on the filtered motion parameters, we characterize recognised visual landmarks as locations on the navigational map. The navigation system can then use references to these landmarks in its driver instructions rather than absolute distances. Experimental results show that landmarks can be located on the navigational map with sufficient accuracy using normal vehicle telemetry and a dashboard camera.**

I. Introduction

Turn-by-turn navigation is probably the most widely-used driving assistance application. With the help of Global Positioning System (GPS) location information and accurate digital maps, drivers are able to reach their destinations while driving through unfamiliar areas. However, current in-vehicle navigation systems can lead to confusion and distraction as drivers try to relate instructions involving distances and street names on the displayed map to their surrounding physical environment, a process which has been compared to assembling a jigsaw puzzle [1]. In fact, humans tend to use landmarks, rather than distances, when constructing spatial cognitive maps [2], and a human-like guidance system that gives navigation instructions in terms of landmark references – "turn right after the post office" rather than "turn right in 100m" – should significantly lower the driver's cognitive load and reduce their navigational errors [3].

We describe a prototype human-like guidance system for driving navigation which uses landmark-based instructions. Instead of using stored landmarks from a map [4], we detect selected landmarks which can be easily recognized in images captured by a dashboard camera. This allows the use of instant visible and other non-map-based landmarks, and may also enable us to bring in dynamic information like: "Follow the yellow car turning left".

In order for the navigation system to perceive and understand the surrounding environment, we employ a variety of computer vision techniques. The landmarks can be detected in real time using a deep learning algorithm as described in [5], after which we need to establish their position on the navigational map. First, the vehicle motion is estimated from a Visual Odometry (VO) algorithm which is tuned to fit the driving scenario. Then an Extended Kalman Filter (EKF) is applied to fuse the VO estimation with multi-sensor data to estimate the vehicle's position and orientation in each frame. Meanwhile, feature points based on recognized landmarks are extracted and tracked between frames. Finally, landmark positions are reconstructed from these feature points using the rectified vehicle motion models, and can therefore be located on the navigation map.

This paper is organized as follows: Section 2 presents an overview of related work. Section 3 details the methodology of the approach. In Section 4, we discuss experimental results and, finally, conclusions and future perspectives are presented in Section 5.

II. Related Work

Despite their popularity, in-vehicle navigation systems have a great deal of room for improvement in order to provide a better driving experience. Landmark-based navigation has the potential to offer more human-like guidance instructions.

There are generally two approaches to landmark-based navigation. The first uses a geographic information system (GIS) [4] where landmarks are stored in a annotated map. Based on the user's location, nearby landmarks, also called Points of Interest (POIs), are presented on the displayed map, and referred to in audio instructions. However, visual information is often crowded on the display, making it hard to read at a glance. Recently, an Apple patent [6] describes referring to restaurants and other landmarks in Siri's turn-by-turn instructions, to make them sound more like directions from a passenger in the vehicle. Despite the increasing availability of annotated POIs, this approach has some limitations. First, this information may easily become outdated, and is usually only available for limited urban areas. Secondly, since the information is decoupled from the current driving environment, it can confuse and frustrate drivers when POIs are invisible or hardly noticeable.

Another approach is to use Computer Vision techniques. Immediate visual information, which is tightly coupled with the driver's perception, offers more flexible and relevant guidance. Robertson and Cipolla's work [7] can accurately estimate the user's location and orientation with a mobile

978-1-5386-1526-3/17/$31.00 ©2017 IEEE

camera by matching the user's view against a pre-stored database. However, maintaining such a database is non-trivial, and querying it may require significant computation and a fast network connection. By contrast, Visual-based Simultaneous Localization and Mapping (V-SLAM) [8], does not rely on a predefined database, since it locates the user in their surrounding environment while building a spatial map at the same time. It is computationally expensive, though, and such systems are generally not suitable for long-distance driving because of the accumulation of errors. A more practical approach, adopted in this work, is mapping the surrounding landmarks from the driver's view onto an accurate navigational map. So far, there has been much work on landmark selection and detection in a real environment [5], [9], but relating the visual perception (landmarks) to a digital map (localization) for navigational purposes has rarely been discussed.

III. Human-like Navigation Guidance Prototype

With the help of deep learning techniques, landmarks of interest can be detected efficiently [5]. Our task is to place these on a navigational map in order to use them as guidance references, which we do in three main stages. First, Visual Odometry (VO) is applied to estimate ego-motion parameters and the vehicle's trajectory. Secondly, an Extended Kalman Filter (EKF) is introduced to correct accumulative errors from the vision-based motion estimation using multi-sensor data. Finally, landmarks are reconstructed and located in the map, based on the filtered vehicle motion parameters.

A. Monocular Visual Odometry

Monocular Visual Odometry [10] uses multiple-view geometry to estimate the position and orientation of the camera/host-vehicle at each instant from a sequence of images. Two problems need to be tackled beforehand: keyframe selection and scale ambiguity.

Keyframe Selection: With images taken from multiple views, the rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$ of camera position $\boldsymbol{P}$ can be estimated from corresponding feature points. It is important, on one hand, to ensure the images contain overlapping areas where sufficiently many feature points can be matched. However, it is impossible to recover the correct 3D position of features points if two views are very close. In a real driving scenario, key frame selection is usually related to the vehicle's dynamics: when the host vehicle is moving quickly, all the frames should be used as keyframes, but when it is nearly stationary, keyframes must be selected with sufficient spacing to maintain a reliable VO estimation.

In our approach, by default, all acquired images are processed as keyframes and the camera motion is estimated between each successive frame. After the inclusion of a new frame, a depth check is applied after the triangulation of feature points. If the median of the reconstructed feature depths exceeds a threshold δ, (which will occur when all the feature points in the new frame are very close to their positions in the previous frame), the changes between the two views are considered to be unlikely to provide accurate motion estimation. Usually it happens when the vehicle is stationary or is only moving slightly. In this case, the current frame is not considered as a keyframe and the vehicle position is not updated for the moment, we call it 'on-hold' stage. Features from the last valid keyframe are kept and tracked through successive frames until obvious vehicle movement is detected. At this point, the frame is labelled as a valid keyframe and the vehicle motion during the 'on-hold' stage is updated by interpolating the estimation between the new keyframe and the previous one. The algorithm continues on this basis.

The threshold δ is obtained by correlation analysis between reconstructed scene distances and vehicle data obtained from a speed sensor. This processing can be done on-line or off-line. At present, we estimate this threshold off-line and use it as a predefined parameter. Fig. 1 shows the correlation of keyframe selection with dynamic vehicle speed, where a keyframe flag equal to 1 means the frame is selected as a keyframe, and 0 means otherwise. As we can see, when the speed of the host vehicle reduces, fewer frames are selected.

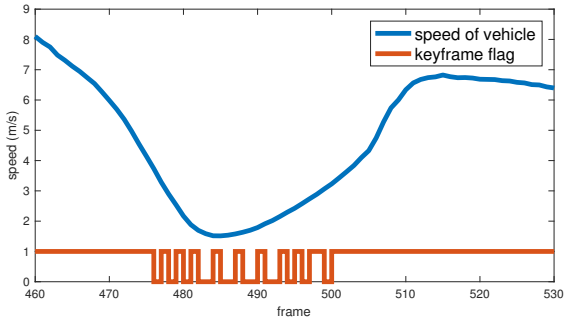

Fig. 1. Correlation of keyframe selection with vehicle speed

Scale determination: In monocular VO, the estimation of the camera motion is limited by scale ambiguity. Different methods have been used to infer the scale relative to real world coordinates, but usually a reference object of a known size is set in the scene to solve the ambiguity. However, this is hard to do in a dynamic driving scenario. Song and Chandraker [11] obtain the scale information by estimating the ground plane and using the height of the camera above the ground. Our dataset, however, provides no accurate information about the camera pose and position, so we infer the scale information from the vehicle's speed using an Inertial Measurement Unit (IMU) which is more precise and allows the algorithm to work with dash cameras which may be relocated. Knowing the speed of the vehicle and the frequency of the video frames, we can calculate the displacement of the host vehicle between two frames. After normalizing the translation vector, this speed per frame is used as the scale in our experiment. Because the frame rate is constant the scale s^v then is noted as the accumulation of speed per frame between two successive keyframes.

Finally, the instantaneous vehicle position estimated by the visual odometry can be represented as:

$$\boldsymbol{P}_k = \boldsymbol{R}\boldsymbol{P}_{k-1} + s^v\boldsymbol{t} \tag{1}$$

if both $k-1$th and kth frame are keyframes, where $\boldsymbol{P}_k$ is the position of camera at kth frame, s^v is the scale as described

above, $\boldsymbol{R}$ and $\boldsymbol{t}$ are respectively the rotation and translation matrix of the camera motion from the previous to the current keyframe.

Algorithm 1 illustrates the tuned monocular VO algorithm.

Algorithm 1 Visual Odometry

$keyframe_1 \leftarrow frame(1)$ first frame;
scale $s^v \leftarrow v_0$ initial vehicle unit speed ;
on-hold stage frame count $n \leftarrow 0$;
for each new frame k with $k > 0$ **do**
    $keyframe_2 \leftarrow$ frame(k);
    detection of features;
    motion estimation $[\boldsymbol{R}|\boldsymbol{t}]$ from features;
    3D reconstruction of feature points Pts;
    if scene depth $d(Pts) < \delta$ **then**
        if $n == 0$ **then**
            position estimation: $\boldsymbol{P}_k \leftarrow \boldsymbol{R}\boldsymbol{P}_{k-1} + s^v\boldsymbol{t}$;
        else
            interpolate $[\boldsymbol{R}|s^v\boldsymbol{t}]$ on frames in on-hold stage;
            position estimation: $\boldsymbol{P}_{k-n+1}..\boldsymbol{P}_k$;
        end if
        update $keyframe_1 \leftarrow keyframe_2$;
        $s^v \leftarrow v_k$;
    else
        on-hold stage frame count: $n = n + 1$;
        scale accumulation: $s^v \leftarrow s^v + v_k$;
    end if
end for

Fig. 2 presents the translation distance of the host-vehicle at each frame from a short trajectory. It compares the translation distance obtained from regular VO estimation with our tuned monocular VO estimation from Algorithm 1. The moving displacement obtained from GPS and IMU are also plotted as reference. As can be seen, the regular VO failed to estimate the vehicle's motion during two segments: the first occurs when the host vehicle is following behind a bus which is making the same turning manoeuvre. In some frames the relative positions of the two vehicles remain constant; the second failing segment is when the host-vehicle is moving very slowly at a speed of around 0.2m/frame (approximately 2m/s). After applying our keyframe selection scheme and interpolation, the motion of the host-vehicle can be estimated continuously and smoothly. (Similar results could be seen in a graph of the vehicle's orientation estimation.) However, as we can see, the tuned VO estimation still presents some deviations from the reference IMU and GPS data.

For navigational purposes, the estimated path from VO should be matched with the map, and the camera motion from VO is used to recover scene geometry. To improve the accuracy, GPS data and IMU data are introduced to fuse with VO result. The camera rotation and translation obtained from Algorithm 1 are used in the fusion process.

B. Data Fusion using Extended Kalman Filter

Camera motion estimation from VO can be inaccurate because of accumulating errors as the driving session proceeds. Optimization solutions such as Bundle Adjustment are often implemented for vision-only odometry applications, but these can be computationally expensive. For our navigation-orientated applications, the natural choice was to correct the localization and motion errors with GPS and IMU data; we do this with Extended Kalman Filtering (EKF) [12].

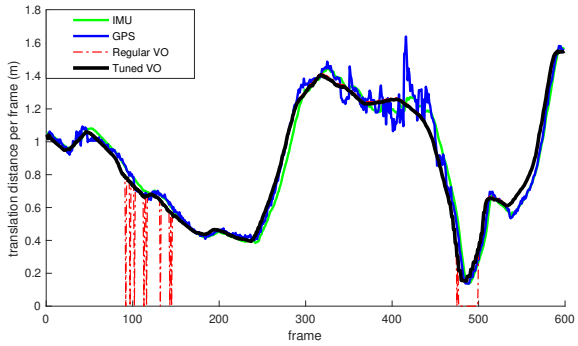

Fig. 2. Comparison of regular/tuned monocular VO on translation estimation

State Model and Measurement: For natural navigation, in addition to the accurate localization of the host vehicle in the map, we need a landmark's position relative to the host vehicle and the planned trajectory. It requires accurate camera pose and scale information at each frame to reconstruct and locate these landmarks correctly. Therefore, in addition to position, the vehicle's orientation and the motion scale should also be rectified with reference sensors. The filtering model is defined as:

$$\boldsymbol{X} = [x, z, \theta, s]^T \tag{2}$$

x, z are the first and third elements in the camera position $\boldsymbol{P}$ in world coordinate, they represent the host vehicle's 2D location on the ground plane, θ is the yaw angle of vehicle motion, s is the scale from Algorithm 1.

Relative, reference sensor observations are used for measurement:

$$\boldsymbol{Z} = [x^{gps}, z^{gps}, \theta^{imu}, s^{gps}]^T + \boldsymbol{\beta} \tag{3}$$

where, x^{gps}, z^{gps} are the GPS-derived location of the vehicle, θ^{imu} is the difference of heading angle estimated from the IMU, s^{gps} is the displacement distance estimated by the GPS per unit time, and $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{W})$ is the measurement noise described in the sensor device's manual. $\boldsymbol{W}$ is the covariance matrix of measurement.

The GPS data is quite noisy as shown in Fig. 2. In the absence of detailed signal strength information from the GPS receiver in the VBOX, we estimate instant GPS accuracy from the number of satellites being tracked. GPS needs at least 4 satellites to provide a precise location; the more satellites, the more precise the localization. We introduced a conditional measurement covariance matrix $\boldsymbol{W'}$: if the number of satellites is less than 4, it is set to a big value; otherwise, it is negatively correlated with the number of satellites within range around $\boldsymbol{W}$. Thus, when the satellite signals are blocked by clustered buildings or dense forest, the navigation system can still work based on VO.

State Prediction and Update: At each step, a prediction is made based on the VO estimation from the last frame. The evolution of the state therefore can be expressed as:

$$f(\boldsymbol{X}_k, \boldsymbol{u}_{k+1}) = \begin{cases} x_{k+1} = x_k + s_k t^x \\ z_{k+1} = z_k + s_k t^z \\ \theta_{k+1} = \phi \\ s_{k+1} = s^v \end{cases}$$

Let $\boldsymbol{u}_{k+1} = [t^x, t^z, \phi, s]^T$ be the input derived from VO. t^x, t^z are the position translations in x and z direction.respectively: $\boldsymbol{t} = [t^x, t^y, t^z]^T$. ϕ is derived from the rotation matrix $\boldsymbol{R}$ of motion estimation, s^v is the same as in Algorithm 1. Taking the model noise into consideration, we have the prediction of the state as:

$$\boldsymbol{X}_{k+1|k} = f(\boldsymbol{X}_k, \boldsymbol{u}_{k+1}) + \boldsymbol{\alpha} \tag{4}$$

where, $\boldsymbol{\alpha} \sim \mathcal{N}(0, \boldsymbol{Q})$ is model noise, and $\boldsymbol{Q}$ is covariance matrix of estimation at frame instant k.

The state is then updated by measurement Z following standard EKF update procedure:

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_{k+1|k} + \boldsymbol{K}_{k+1}(\boldsymbol{Z}_{k+1} - \boldsymbol{X}_{k+1|k}) \tag{5}$$

where, $\boldsymbol{K}_{k+1}$ is Kalman gain for each step.

After data fusion using EKF, accumulating errors from the VO can be avoided. The rectified state parameters are also ready to be used for landmark localization.

C. Landmark Localization

When a navigational instruction needs to be delivered, we want the navigation system to give human-like guidance based on landmarks, since drivers often have a poor intuitive understanding of numerical distances. For this reason, landmarks must be located on the navigation map in order to be used as references to the planning path.

For each frame, the landmark detection process can be executed in parallel with the VO. Most existing object-detection methods present their results in bounding-boxes. For example, Wiles *et al.* [5] detect the outline of selected landmarks such as bus stops, corner shops, etc.

We extract SIFT feature points from each landmark's bounding-box and track them through subsequent frames. Their position relative to the host vehicle is reconstructed using the rectified camera motion from the EKF: the rotation and translation used for triangulation are rectified using updated yaw angle θ and scale s from the data fusion processing. The average position of the feature points in each bounding box represents the position of the landmark. Finally, by relating this to the rectified host-vehicle position x, z, we can easily locate the landmarks in the navigational map. The feature tracking is done once a landmark is detected; the landmark location is reconstructed and updated after each frame.

The navigation system can now use landmarks as references to deliver human-like instructions: a process which will form the next phase of this work.

IV. Experiment Design and Results

Experiments were conducted using multi-sensor data collected from real driving scenarios in a natural environment, to verify the proposed approach. The data is collected from a dash camera and a VBOX data logger installed inside the host vehicle, and it was selected primarily from travel in built-up areas and on city roads, since these scenarios generally contain more meaningful landmarks. The experiments consist of two parts: first, a 2km long trajectory segment in urban area is analyzed along which different type of landmarks are presented. Second, a collection of map included landmarks (i.e. bus stops) at different locations is analyzed and compared with Open Street Map labellings.

A. Data Acquisition and Correction

GPS and IMU data, including host-vehicle dynamics and locations, were collected by a VBOX data logger at a frequency of 10Hz. A monocular camera installed behind the windshield recorded the front view video at 30fps, with a resolution of 1280×720 pixels. To match the frequency of GPS, the frames are extracted from the video at frame rate of 10fps. All sensor data are time-synchronized before the experiment. Since differential GPS is not enabled in this dataset, we consider it as a reference, not as reliable ground truth.

Fig. 3. Example frame with labelled landmark: a traffic sign

One challenge about this dataset is that the camera pose respect to the host-vehicle is not strictly regulated. As shown in Fig. 3, the camera orientation is not aligned with the vehicle's orientation, it means that the estimated camera motion from VO does not represent the real motion state of the host-vehicle. Fortunately, the GPS and navigation map indicate when the host vehicle is moving straight forward on a planar road. Comparing this trajectory segment with the estimated camera movement path from VO, we can get an approximate yaw angle of the camera pose with respect to the host-vehicle coordinates. In addition, for this dataset, the roll angle of camera relative to the host-vehicle can also be inferred by extracting the edge of windshield. (Another more general method that can be used to solve this problem involves estimating the ground plane in camera coordinates). However, the pitch angle of the camera is hard to estimate since the car is vibrating all the time. In our data analysis, no solid

TABLE I
ESTIMATION ERROR OF WITH REFERENCE DATA ACCURACY

Table	Position(m)	Translation	Yaw(°/m)
VO Err.	1.58	3.85%	0.0029
EKF Err.	0.17	0.79%	0.0015
Ref. Acc.	3.00(m)	/	0.01(°)

evidence indicates a specific pitch angle, so we assume a pitch angle of zero during the experiment. All pose angles are considered with estimation errors. The VO estimation can then be transformed to represent the vehicle's movement. Despite the challenge, an ability to cope with these variations rather than assuming a fixed camera location in the vehicle has the benefit that portable visual navigation devices could also be used for this kind of natural navigation.

B. Experimental Results

First, a driving trajectory of 2km is selected to evaluate both vision based and EKF fused odometry estimations. The evaluation metrics are: the average differences of the estimated position(m), translation error [13] and yaw angle error(°/m). The yaw angle error is a variant version of rotation error in [13]. Table I demonstrates that the fused odometry estimation is more precise than the tuned VO algorithm. Bottom row of Table I lists the accuracy of the reference data which is obtained from the device's manual, that provides a relative basis for the estimation accuracy.

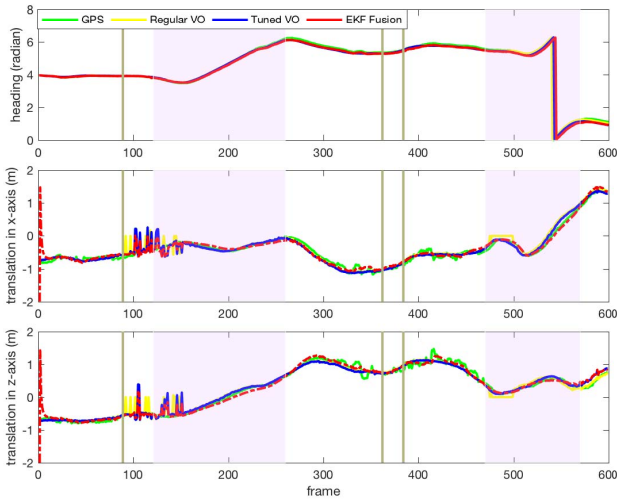

Fig. 4. Plots of heading and translation of host vehicle at each frame

We pick a representative segmentation of 500 metres from this trajectory for detailed analysis. It contains two roundabouts; with a brief pause before entering the second one. Fig. 4 presents the host-vehicle's estimated heading direction and translation in x-axis and z-axis along frame stamps. The three vertical lines indicates when the landmarks are detected and start being tracked. The half-transparent pink blocks indicates the duration of two roundabouts. We compare the estimations from regular VO, tuned VO, EKF-based data fusion, and GPS data are also listed as references. The top sub-figure shows the heading plots from different algorithms, and they tend to have very small deviation, but after every roundabout, the deviation becomes more visible.

The middle and the bottom sub-figures illustrate the translation deviation from the GPS reference. Mostly, the estimated translation is relatively close to the reference. However, there is a major deviation when entering the first roundabout. Compared to regular VO, the tuned VO effectively reduced the duration of deviation, but cannot remove it completely. The reason is that a moving bus occupies the major view of the dash camera during that period, leading to unreliable ego-motion estimation.

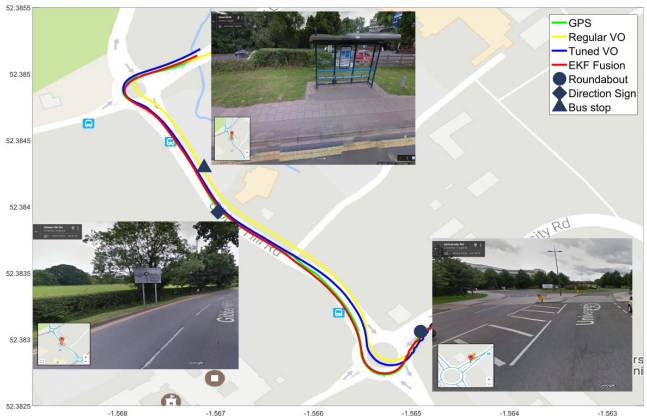

Fig. 5. Example of landmark localization result

Fig. 5 plots the estimated trajectories from tuned VO and EKF-based data fusion, GPS trajectory is set as reference. As expected from previous analyses, a major trajectory deviation appears during the first roundabout using tuned VO based estimation, and minor deviations accumulate along the driving path. Data fusion using EKF, however, effectively reduced the estimation deviation from the reference trajectory.

Knowing where the host-vehicle is located and how it moves, we can reconstruct landmarks accordingly. Three example landmarks are selected manually along the trajectory in this segment, while the host vehicle is moving at different speeds and in different orientations: a traffic sign when entering a roundabout, a direction sign, and a bus stop. Bounding boxes on the landmarks were labelled manually in the frame when the host vehicle is approaching from 50m away. Detected SIFT features from these bounding boxes are then tracked and triangulated during the next 2 seconds (20 frames). Means of the estimated landmark locations are marked on the map in Fig. 5. However, no ground truth is available, so we used Google Street View to check whether the landmarks can be observed at each estimated location. Naturally, there are discrepancies because images in Google Street View are captured at discrete intervals. Still all of the landmarks are successfully found near their estimated location. In additional, we have localized 5 traffic lights from other trajectory segments. Despite the lack of ground truth, the standard deviation of the landmarks' location estimation is quite small, as indicated in Table II. It illustrates that our method can provide consistent and stable estimation of landmarks' location with only brief visibility.

TABLE II
STANDARD DEVIATION OF ESTIMATED LANDMARK LOCATIONS

Landmark	Roundabout	Direction sign	Bus stop	5 Traffic lights
σ_x(m)	2.63	0.93	0.81	7.76
σ_z(m)	1.95	0.71	1.02	2.94

For the second part of experiment, we tried to localize POI landmarks found in Open Street Map (OSM) using our algorithm and compare the result with their location on the map. During the data collection, we noticed that not all interesting landmarks are available in OSM, for example, traffic lights in branch roads. In addition, the landmarks labelled as POIs are not always visible from the dash camera. In the end, we selected five bus stops for comparison. The result is shown in Table III. As can be seen, the standard deviations for VO-based localizations from different frames are mostly about 5m, except for bus stop number 3 (BS3). The reason for such a big deviation here could be a combination of the inaccurate ego-motion estimation and the GPS data, since the GPS localization in this segment is very noisy. Situation like this is difficult to avoid, we are hoping to improve the result by adding filters for the landmarks as well in the future.

By contrast, the distance between our VO-localized landmarks and their positions in OSM have low estimation deviation. Bus Stop number 1 (BS1) shows the biggest difference from its OSM location. We cross-checked the position of this landmark against Google Maps, and it appears that the accuracy of some of the POI map locations in OSM is decidedly weak. A robust general evaluation method of the localization accuracy therefore remains to be found.

TABLE III
LANDMARKS LOCALIZATION AGAINST OSM

Landmark	BS1	BS2	BS3	BS4	BS5
σ_x(m)	5.27	1.02	23.81	1.75	0.87
σ_z(m)	0.57	1.43	13.74	1.72	1.21
d_{OSM}(m)	19.22	10.24	16.11	4.27	8.93

Despite this, the early experimental results from our method suggest that it provides a promising prototype algorithm for locating landmarks for human-like navigation guidance.

Our local processing platform is a standard PC running Mac OS on a 2.66GHz Intel CPU. The computation environment is MATLAB R2016b. The average run-time for EKF-based host-vehicle localization and mapping is around 544ms per frame. For landmark localization it costs 9.5ms per frame. Since the experiments were done purely for prototype development, it is likely that the algorithm would be able to run in real-time after optimization.

V. CONCLUSION AND FUTURE WORK

We have presented a first step towards the development of a natural, human-like navigation system using Computer Vision techniques. We have demonstrated that the fusion of VO with other sensor data effectively assists in converting a visible landmark in the driver's view to a position on a navigational map. However, a full qualitative evaluation remains difficult in the absence of reliable ground truth data.

The main contributions of this paper are:

- A tuned VO algorithm which is suitable for driving navigation
- A landmark-localization-orientated EKF filter for multi-sensor data fusion
- Demonstrating the feasibility of landmark-based navigation.

Our next step will be to combine our current work with a routing algorithm to build a complete human-like guidance system. In the short term, we can approach a pure vision-based navigation system by using landmarks for partial bundle adjustment to improve the accuracy of Monocular VO. Additionally, we are aiming to build a more reliable landmark dataset from POIs stored in Open Street Map, which will form a reference for detection training and localization evaluation.

ACKNOWLEDGMENT

The work presented in this paper was funded and supported by Jaguar Land Rover, Coventry, UK.

REFERENCES

[1] Barry Brown and Eric Laurier. The normal natural troubles of driving with GPS. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1621–1630. ACM, 2012.

[2] "Human-Like Local Navigation System Inspired by a Hippocampal Memory Mechanism" *Brain-Inspired Information Technology*,pp.29–32,2010.

[3] Morgane Roger, Nathalie Bonnardel, and Ludovic Le Bigot. Landmarks use in speech map navigation tasks. *Journal of Environmental Psychology*, 31(2):192–199, 2011.

[4] Markus Dräger and Alexander Koller. Generation of landmark-based navigation instructions from open-source data. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*,EACL'12 pages 757–766, 2012.

[5] Olivia Wiles, Marwa Mahmoud et al.: "Towards a User-Centric In-Vehicle Navigational System" *AutomotiveUI 16*. In *Proceedings of AutomotiveUI 16*, ACM, 2016.

[6] A.K. Kandangath and X. Tu. Humanized navigation instructions for mapping applications, April 23 2015. US Patent App. 14/061,208.

[7] Duncan P Robertson and Roberto Cipolla. An image-based system for urban navigation. In *Proceedings of the 15th British Machine Vision Conference (BMVC'04)*, volume 2, pages 819–828, 2004.

[8] Ayoung Kim and Ryan M Eustice. Perception-driven navigation: Active visual slam for robotic area coverage. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3196–3203. IEEE, 2013.

[9] Bassel Zeidan, Sakyasingha Dasgupta, Florentin Wörgötter, and Poramate Manoonpong. Adaptive landmark-based navigation system using learning techniques. In *International Conference on Simulation of Adaptive Behavior*, pages 121–131. Springer, 2014.

[10] Friedrich Fraundorfer and Davide Scaramuzza. Visual odometry: Part ii: Matching, robustness, optimization, and applications. *IEEE Robotics & Automation Magazine*, 19(2):78–90, 2012.

[11] Shiyu Song and Manmohan Chandraker. Robust scale estimation in real-time monocular sfm for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1566–1573, 2014.

[12] Lijun Wei, Cindy Cappelle, Yassine Ruichek, and Frédérick Zann. Intelligent vehicle localization in urban environments using ekf-based visual odometry and gps fusion. *IFAC Proceedings Volumes*, 44(1):13776–13781, 2011.

[13] Andreas Geiger and Philip Lenz and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.