https://doi.org/10.2352/ISSN.2470-1173.2018.17.AVM-281

Visual SLAM and Localization – The Hard Cases

Catherine Enright, Valeo Detection Systems, Tuam, Ireland;
Bassam Abdallah Comfort and Driving Assistance Systems - Driving Assistance Research, Bobigny, France.

Abstract

We present a visual SLAM pipeline that is efficient, robust and accurate. It is applied to the trained parking use case. In this case the SLAM algorithm builds a "trained" map on the first pass, typically driven by the driver. In subsequent passes the algorithm localizes to the trajectory, thus allowing the vehicle to autonomously follow the trained path.

A visual SLAM system for autonomous vehicles is an attractive option as it utilizes relatively cheap sensors that are typically already mounted on the vehicle for other tasks. However using a visual SLAM approach has challenges, in this paper we specifically look at the localization task in difficult cases.

The system is designed to operate in an uncontrolled environment. Between map generation and localization there may be significant changes, different dynamic objects, missing structure, moved structure or the scene may be visually different due to illumination changes or changing weather conditions. These are the so called hard cases.

We present an approach, which runs in real-time, designed to tackle the hard cases. The approach has been evaluated both at the bench and in-car.

Introduction

SLAM (Simultaneous Localisation and Mapping) algorithms have been the subject of extensive research since the foundations were laid by Smith and Cheeseman in 1986 in their work on the representation and estimation of spatial uncertainty [1]. SLAM itself was first conceived in the early '90s by Hugh Durrant-Whyte and Leonard [2] but it took another decade before camera based SLAM algorithms were investigated. Initial approaches used stereo cameras, monocular camera systems came a bit later and many of the techniques used came from parallel research in the vision community on the Structure for Motion task.

Now there are a variety of Visual SLAM approaches to choose from i.e. feature-based (e.g. ORB-SLAM [3]), direct methods which can be either dense (e.g. DTAM [4]) or semi-dense (e.g. LSD-SLAM [5]). CNNs are now being applied to the problem with CNN-SLAM [6] a notable example. Object-based or semantic SLAM is also an active area of research.

Despite the extensive research real world outdoor applications are still scarce. Outdoor scenes present particular challenges for visual SLAM algorithms – they are uncontrolled environments with significant variations in illumination, weather and scene structure.

As noted by Fuentes-Pacheco et al. [7] many visual SLAM approaches fail under the following conditions:

- in external environments,
- in dynamic environments,
- in environments with too many or very few salient features,
- in large scale environments
- during erratic movements of the camera
- when partial or total occlusions of the sensor occur.

The key to a successful visual SLAM system that can be deployed in an autonomous vehicle is the ability to operate correctly despite these difficulties.

In this paper we look at the specific use case of Visual SLAM, that of trained parking in automotive vehicles. In this use case the SLAM algorithm builds a "trained" map on the first pass, typically driven by the driver. In subsequent passes the algorithm localizes to the path, thus allowing the vehicle to autonomously follow the trained path.

The approach is chosen to specifically deal with the uncontrolled outdoor environment. Between map generation and localization there can be significant changes. Dynamic objects like pedestrians or other vehicles can be present during training but absent in replay. Different dynamic objects may be present in the scene during the localization phase. Static structure in the scene may have moved or be different e.g. in a typical home scenario refuse bins may be in a different location on different days. Illumination changes and weather variation by far present the greatest challenge for a vision based SLAM system operating in the outdoors with day to night a particular challenge.

System Requirements

For many SLAM algorithms the focus is on creating an accurate representation/model of the scene. In the trained parking use case the focus is slightly different; here the most important function of the algorithm is that it can accurately localize to the trained scene.

Four fish-eye cameras provide an omni-directional surround view, as shown in Figure 1, this is the perception input.

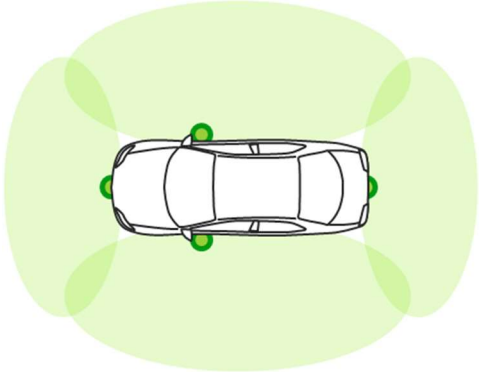

Figure 1: Vehicle equipped with Valeo SVS cameras

The algorithm is required to run in real-time while the vehicle is traveling at a velocity of 10kph. The memory required for the

trained map must be minimal (of the order of few Mb per camera for a ~20m trajectory). Although the tests included in this paper are undertaken on a single core on a PC, the system is designed to be deployed on an embedded platform.

Robust Re-localisation Methodology

As re-localisation success and accuracy are the key performance indicators for the use case in question a feature-based Visual SLAM approach was chosen. The proposed solution has 2 main modes of operation – Training and Replay.

Training Mode

The aim of the training phase is to create a sparse consistent map of trained trajectory points and trained features. Trained trajectory points are key locations that together identify the path followed by the vehicle in training.

Trained features represent a set of unique 3D positions in the world associated to single/multiple 2D visual feature(s). A visual feature is made of 2D coordinates and a visual descriptor. The descriptors are such that they uniquely define a feature and they can be associated with the features extracted in a subsequent replay. The system is designed to work with any feature descriptor but a sufficiently robust descriptor is recommended. The feature descriptor storage during training and matching during replay is the key enabler for the vehicle to localise itself during replay against the training data.

The concept of "key frames" and "normal frames" is important for this solution. During training both frame types are processed but only the trained trajectory points and trained features associated with key frames are saved in the trained map and used for replay.

Frames are bundled into windows with a key frame at the start followed by normal frames. Key frames are dynamically selected based on a combination of distance travelled and the number of features matched to previous key frame.

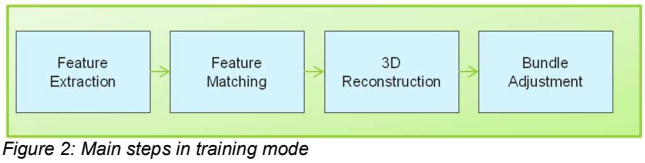

Figure 2: Main steps in training mode

In the training phase, a set of visual features are extracted from each live camera frame, a visual descriptor is stored for each feature. The number of features extracted is limited to a fixed number to ensure an efficient run-time is achieved.

Visual features are matched according to their descriptor to subsequent frames. Using these matches a 3D reconstruction is performed to estimate a triangulation for each feature, in a 3D world coordinate system. The 3D reconstruction also provides an estimate of the vehicle motion, via the estimation of the essential/fundamental matrix.

Each window consisting of a key frame and subsequent normal frames is bundle adjusted to give the optimal trajectory positions and 3D feature positions. The bundle adjustment step involves a non-linear optimisation where the reprojection error of the 3D features is minimised. Namely the trained trajectory $\{\tilde{p}_j\}$ and trained features $\{\tilde{t}_i\}$ are obtained via the following minimization:

$$\{\tilde{p}_j\},\{\tilde{t}_i\} = \underset{\{p_j\},\{t_i\}}{\operatorname{argmin}} \sum_{i=1}^{n} \sigma\left(\sum_{j=1}^{m} \delta_{ij} \left\| \prod(p_j, t_i) - o_{ij} \right\|^2\right)$$

Where o_{ij} is the 2D observation of a trained feature i on the trajectory point j. The quantity p_j (resp. t_i) represents the estimated camera position (resp. triangulated position of the landmark i) obtained after 3D reconstruction. The operator $\prod(p_j, t_i)$ stands for the projection function of the 3D estimate on the j-th image plane. δ_{ij} is a convenient term for the formula (it is implicit in the implementation) that ensures the landmark i was observed on at least two frames among which frame j. Finally σ refers to a loss function allowing this equation to gain robustness towards outliers.

Replay Mode

In the replay phase, a new set of live visual features are extracted from each processed frame, as during training. As shown in Figure 3 there are two main steps in replay; topological re-localisation and metric re-localisation. They can be pictured as a cascaded re-localisation process where the position of the vehicle is sequentially refined.

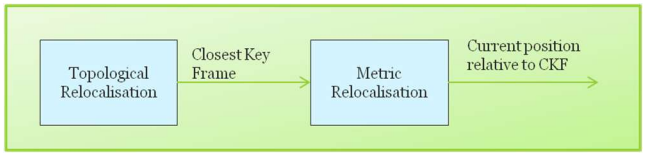

Figure 3: Replay pipeline

For the initial topological localisation (i.e. lost robot problem) a subset of the new features is selected and the complete trained cloud is searched to find matching descriptors. The key frame with the most matches is selected as the "closest key frame" and then an attempt is made to match the complete set of new features to the features in the "closest key frame". This technique is accurate and works well when the replay scene is similar to the trained trajectory. However when the replay scene is significantly different due to illumination/scene changes the response rate of this method falls to 0. To overcome this issue another algorithm takes place in case of failure of the first one. Namely the closest key frame is selected based on a frame signature which similar concept to the *holistic feature vector* used by Lategahn et al. [8] for loop closure detection. Each frame is divided into a 4x4 grid and the summary descriptions of each cell are combined to provide the signature. This method always returns a proposal for the closest key-frame, a subsequent check is thus included to validate the returned key frame.

Once the closest key frame is determined, the metric localisation can be attempted. Live visual features (o_i) from the live frame are matched to a subset of trained visual features as including those observed in the closest key frame and its neighbours during training. If a sufficient number are matched the current vehicle position ($\tilde{p}$) relative to the trained trajectory is

determined using a least squares optimisation. The live visual features matched with trained visual features are associated to the trained features $\tilde{t}_i$. The latter are re-projected into the current live frame and the re-projection error is calculated as the difference between the re-projected (by means of $\prod(p, t_i)$) position and the observed position (o_i) in the current frame. This error is minimised by optimising the vehicle position (p) relative to the trained trajectory. Strict filtering of outliers with a loss function (σ as for train) ensures a reliable position $\tilde{p}$ as:

$$\tilde{p} = \underset{p}{\operatorname{argmin}} \sum_{i=1}^{n} \sigma\left(\delta_i \left\| \prod(p, \tilde{t}_i) - o_i \right\|^2\right)$$

On subsequent frames the search is restricted to the "closest key frame" and its neighbours. If however a sufficient number of matches are not made within this set the complete trajectory is once again searched.

Multi-Camera

As shown in Figure 1 SVS cameras allow us perceive the environment, build our map and relocalise to the map. By using multiple cameras a much more robust localization can be achieved. Features are not matched between camera views; instead the optimization step considers the optimization of the vehicle pose based on observations from all cameras.

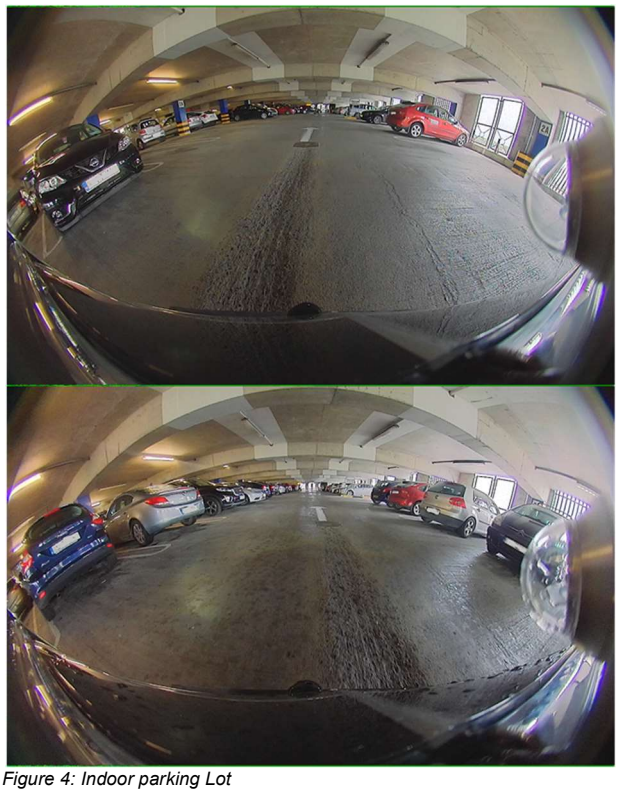

Figure 4: Indoor parking Lot

The Hard cases

Indoor Public Parking Lot

Figure 4 below shows a typical indoor parking scene. The top frame was taken from a video captured in the morning when the car park was relatively empty. By mid-afternoon the scene had changed significantly. Most of the cars that were present in the morning were gone in the afternoon. Many new cars were added to the scene obscuring features that could be observed in the morning. The roof does not change but unfortunately it is a repeated structure making localisation to unique position difficult. It can also be observed that the vehicle starting position is slightly different in each scene, thus features are observed from a slightly different rotation.

The fact that our trained map is very sparse adds to the challenge. Because of the requirement to be able to run in real-time on an embedded platform we are limited in the landmarks we can store. If most of the scene is obscured or changed matching between training and replay is difficult.

This is a scene that highlights the main advantage of a multi-camera system. With only one camera the algorithm fails to reliably localize on all frames. However the introduction of a second camera view results in a reliable and accurate localization from start to end.

Figure 5: Trained trajectory overlaid on the replay scene.

Day Night

Figure 6 illustrates the challenge of illumination change. Training was carried out in on a bright day and replay at night. The

structure of the scene remains unchanged between training and replay. The same cars, trees and refuse bins can be observed however they appear visually different due to the change in lighting conditions.

Figure 6: Illumination changes between training and replay

Careful use of a robust illumination invariant feature detector/descriptor enables replay in most of this scene. The key however is a good topological re-localisation – course localization to the correct trained key frame is vital.

A second challenge in this scene is the distance of the objects in the scene from the vehicle, the trees being the obvious example. Trees are particularly interesting as on windy days they are not stationary and, when added to the trained map, can therefore lead to noisy localization results. Once again this is where using multiple cameras improves results. The re-localisation success rate with front camera only is 96.38% but with front and rear camera it rises to 100%. What we also note is that using the same number of features spread over 2 cameras results in more accurate smoother results.

There are however cases where VSLAM does not succeed. In very dark featureless environments the algorithm cannot even train sufficiently well as it is not possible to match enough features between frames. An example is shown in Figure 7 where the illumination in the scene was less than 10Lux.

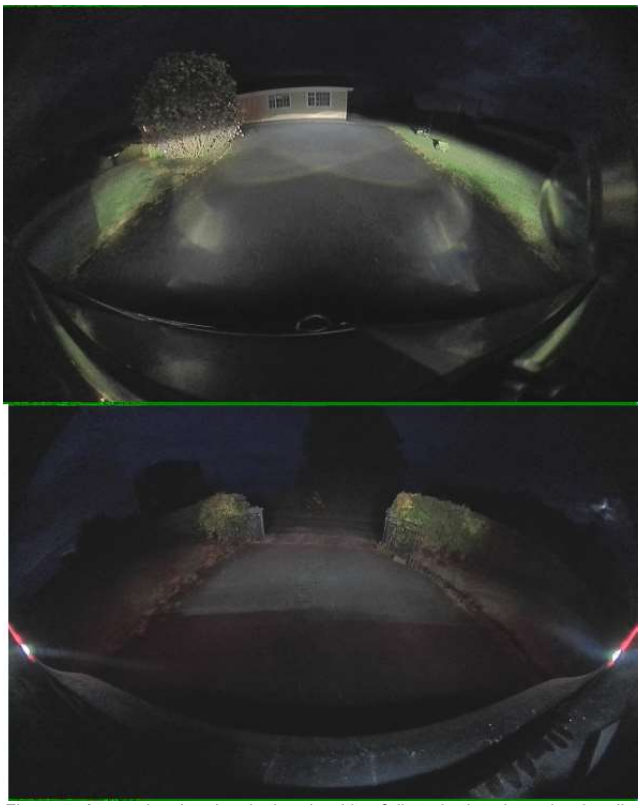

Figure 7: At very low Lux levels the algorithm fails to both train and re-localise

Revisiting a scene after 10 months

In this case we revisit a scene after a 10 month gap i.e., training was performed on a capture from December 2016 and replay was performed on a capture from October 2017.

It can be noted, in Figure 8 that the lighting conditions are very different with low sunlight illuminating the training scene and replay attempted on an overcast day. Also to be noted are the changes in vegetation between the seasons. The replay starts approximately 6 metres from where the training started and there are both lateral and rotation offsets.

Figure 8: Scene revisited after 10 months

In this scenario with 2 cameras active we re-localise successfully on 73.7% of the frames.

Conclusions

A feature-based visual SLAM based solution can effectively meet the challenging requirements of the trained parking use case.

Dispersing the observed landmarks around the vehicle in a multi-camera solution significantly improves the algorithm, making it more robust to structure change, illumination change, glare and occlusion. Using just a wide-angle front and rear view is sufficient. While the addition of the wing-mirror cameras also increases accuracy and robustness the improvement is not enough to justify the extra memory and run-time. Capturing front and rear view has the additional advantage that localisation to the trained route can be done in either direction (i.e. by matching trained features from the front view camera to live features from the rear view camera and vice versa)

The results discussed in this paper are solely from the visual SLAM algorithm. In the full trained parking pipeline a downstream Kalman filter is used to fuse the results with mechanical odometry. This has the effect of smoothing the replay trajectory and elegantly handling noise or inaccurate localisations.

The algorithm is efficient and can run in real-time. The memory requirements of the trained map are kept to a minimal because it is a feature-based visual SLAM approach. The algorithm is robust to the significant scene changes that can occur over time in an outdoor environment.

Acknowledgement

The authors would like to acknowledge the hard work and dedication of the VSLAM team in Valeo.

References

[1] R. C. Smith and P. Cheeeman, "On the representation and estimation of spatial uncertainty.," *The international journal of Robotics Research,* vol. 5, no. 4, pp. 56-68, 1986.

[2] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localisation for an autonomous mobile robot," in *In Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on (pp. 1442-1447)*, 1991.

[3] R. Mur-Artal, J. M. M. Montiel and J. D. Tardos., "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics,* vol. 31, no. 5, pp. 1147-1163, 2015.

[4] R. A. Newcombe, S. J. Lovegrove and A. J. Davison, "DTAM: Dense tracking and mapping in real-time.," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011.

[5] J. Engel, T. Schöps and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision*, 2014.

[6] K. Tateno, F. Tombari, I. Laina and N. Navab, "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction," in *arXiv preprint arXiv:1704.03489*, 2017.

[7] J. Fuentes-Pacheco, J. Ruiz-Ascencio and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: a survey," *Artificial Intelligence Review,* vol. 43, no. 1, pp. 55-81, 2015.

[8] H. Lategahn, Mapping and Localization in Urban Environments Using Cameras., KIT Scientific Publishing, 2014.