Visual Odometry and Place Recognition Fusion for Vehicle Position Tracking in Urban Environments

Safa Ouerghi, Rémi Boutteau, Xavier Savatier, Fethi Tlili

sensors

MDPI

Article

Visual Odometry and Place Recognition Fusion for Vehicle Position Tracking in Urban Environments

Safa Ouerghi [1,*], Rémi Boutteau [2], Xavier Savatier [2] and Fethi Tlili [1]

[1] Carthage University, SUP'COM, GRESCOM, El Ghazela 2083, Tunisia; fethi.tlili@supcom.tn
[2] Normandie University, UNIROUEN, ESIGELEC, IRSEEM, 76000 Rouen, France; remi.boutteau@esigelec.fr (R.B.); xavier.savatier@esigelec.fr (X.S.)
* Correspondence: safa.ouerghi@supcom.tn

Received: 16 February 2018; Accepted: 19 March 2018; Published: 22 March 2018

Abstract: In this paper, we address the problem of vehicle localization in urban environments. We rely on visual odometry, calculating the incremental motion, to track the position of the vehicle and on place recognition to correct the accumulated drift of visual odometry, whenever a location is recognized. The algorithm used as a place recognition module is SeqSLAM, addressing challenging environments and achieving quite remarkable results. Specifically, we perform the long-term navigation of a vehicle based on the fusion of visual odometry and SeqSLAM. The template library for this latter is created online using navigation information from the visual odometry module. That is, when a location is recognized, the corresponding information is used as an observation of the filter. The fusion is done using the EKF and the UKF, the well-known nonlinear state estimation methods, to assess the superior alternative. The algorithm is evaluated using the KITTI dataset and the results show the reduction of the navigation errors by loop-closure detection. The overall position error of visual odometery with SeqSLAM is 0.22% of the trajectory, which is much smaller than the navigation errors of visual odometery alone 0.45%. In addition, despite the superiority of the UKF in a variety of estimation problems, our results indicate that the UKF performs as efficiently as the EKF at the expense of an additional computational overhead. This leads to the conclusion that the EKF is a better choice for fusing visual odometry and SeqSlam in a long-term navigation context.

Keywords: real-time navigation; visual-odometry; SeqSLAM; loop-closure; EKF; UKF

1. Introduction

Autonomous vehicles have recently received great attention in the robotics, intelligent transportation, and artificial intelligence communities. Accurate estimation of a vehicle's location is a key capability to realizing autonomous operation. Currently, the leading technology in this setting is GPS receivers to estimate their absolute, georeferenced pose. However, most commercial GPS systems suffer from limited precision and are sensitive to multipath effects (e.g., in the so-called "urban canyons" formed by tall buildings), which can introduce significant biases that are difficult to detect in addition to sometimes being unavailable (e.g., in tunnels). To provide alternatives to GPS localization, many sensory devices have been applied to carry out the localization task, including visual sensors that are often desirable due to their low-cost, wide availability and passive nature.

Vision-based localization techniques fall into several broad categories including real-time Structure from motion (SfM) or Visual Odometry (VO) and Place Recognition. While VO methods calculate the egomotion by incrementally estimating the rotation and translation undergone by the vehicle using only the input of a single or multiple cameras, Place Recognition methods are based on learning a database of images (the map) and the vehicle consecutively tries to find matchings between this database and the actual visual input (query image(s)). Within the place recognition

paradigm, numerous research papers have addressed visual appearance-based place recognition by ground vehicles especially in varying environments [1–5]. A leading method is SeqSLAM aiming at matching image sequences under strong seasonal and illumination changes through the computation of image-by-image dissimilarity scores between all query and database images. Despite its performance under strong changes, SeqSLAM tends to fail in correctly matching images when dealing with changes in viewpoint that can be found between them [6].

Hence, our motivation is to create a vision-driven localization method in urban environments that uses cues from SeqSLAM and VO. Our method is able to localize vehicles in the global reference system if a georeferenced database of images has been learned beforehand. However, in this paper, we address the problem of long-term navigation where loop closures, referring to when the vehicle has returned to a past location after having discovered new terrain for a while, are used to reduce the drift caused by VO. Such detection makes it possible to increase the precision of the actual pose estimate. To this end, we present the integration of the robust sequence-based recognition capabilities of the SeqSLAM system with the accurate 3D metric properties of monocular VO in a probabilistic framework through the use of the Extended Kalman Filter (EKF), the efficient recursive estimator [7] and the unscented Kalman filter (UKF). Keeping power-, space- and weight-constrained applications in mind, we prefer to avoid additional sensors and to utilize only visual cues available in monocular sequences. In fact, despite years of research, monocular-based-localization systems are still an exciting open problem.

The paper proceeds as follows: Section 2 presents some related work. Section 3 summarises the SeqSLAM's main components and its improved version. Then, our approach details including the position tracking, the uncertainty estimation and the fusion through the filtering techniques are presented in Section 4. Finally, Section 5 presents the results, discusses the outcomes of this paper and presents suggestions for future work.

2. Related Work

Several approaches have been developed in the past few years, which frame self-localization as a retrieval task. Towards this goal, multiple representations of the world have been adopted, namely, visual Bag-of-Words (BoW), visual features and 3D point clouds. BoW representation has been first used by the popular algorithm FAB-MAP [8] that relies on extracting scale-invariant image keypoints and descriptors. The descriptor vectors are then quantized using a dictionary trained on prior data. Fab-Map achieves robust image recall performance for outdoor image sequences up to 1000 km [9]. Other approaches dealing with visual features either use a trained georeferenced map of simple visual 3D features [10], or visual features from 3D building geometry [11]. In the case of 3D point clouds, the localization is performed by retrieving point clouds which are similar to the current scene [12].

However, the success of visual features-based approaches is very dependent on the quality of the visual vocabulary, and in turn on the prior data and the reliability of extracting the same visual keypoints and descriptors in images with similar viewpoints. The latter is particularly problematic when there are large lighting variations and scene appearance changes due either to seasonal changes or to day and night cycles. Significant performance improvements over Fab-MAP under extreme lighting and atmospheric variations have been achieved within the SeqSLAM system, which relies on the use of a whole image regardless of its content by searching in a pre-learnt database of images for the most similar sequence to the query sequence [1,2]. SeqSLAM proved its success with very low resolution imagery [2,13], on large-scale environments of 3000 km over four seasons [6], with UV imagery [4] and with ConvNet features [5]. However, one of SeqSLAM's most significant drawbacks is its lack of viewpoint invariance inherited from the use of global image matching.

On the other hand, when the initial position is known, self-localization is achieved by visual odometry that yields relative motion estimates, integrated to obtain an estimate of the vehicle's current position. High accuracy has been achieved using stereo-based SfM systems [14]. Good performance of monocular systems has also been demonstrated on the KITTI visual odometry benchmark [15],

for example in [16,17], but the incremental nature of these methods inevitably leads to drift. Simultaneous Localization and Mapping (SLAM) methods attempt to reduce this drift by using landmarks and jointly optimizing over all or a selection of poses and landmarks [18,19]. Drift can be further reduced by revisiting places several times and detecting loop closures in the traveled trajectory [20].

However, SLAM methods suffer from issues in terms of speed and map size which limit their application at large scales. Even though efficient optimization strategies using either incremental sparse matrix factorization [21] or relative representations [22] have been used, several recent works have instead relied on publicly available maps [23,24], road networks [25] or satellite images [26] to address the localization of ground vehicles.

Other approaches are based on the fusion of two or multiple algorithms. In fact, data from different algorithms might be combined to derive a more accurate estimate of the vehicle's pose, as their uncertainties might be complementary. Illustrated in [27] is the fusion of visual odometry, used to track the position of a camera-equipped Micro Aerial Vehicle (MAV) flying in urban streets with an air-ground image matching algorithm using a cadastral 3D city model by means of a Kalman filter. Additionally, our approach is based on the fusion of VO and the appearance-based place recognition algorithm SeqSLAM. We adapted SeqSLAM for use in the context of long-term navigation for loop closure detection in order to reduce the drift caused by VO. The database of images is, therefore, created online. However, in the case of learning a geo-referenced database beforehand, the method remains valid and more accurate estimates could be obtained in the global reference frame without assuming a loopy trajectory.

3. Appearance-Based Global Positioning System

In this section, we briefly present the main components of the state-of-the-art appearance-based global positioning algorithm SeqSLAM and then show some components of the Sequence Matching Across Route Traversals (SMART) system, built upon SeqSLAM to overcome some of its limitations. The SeqSLAM algorithm, using a simple whole-image comparison with Sum of Absolute Differences (SAD), demonstrated impressive place recognition performance across significant condition variance such as seasonal changes and day to night transitions where feature-based methods failed entirely and in the case of low quality imagery (low resolution, low depth, and image blur). In order to increase the discriminative nature of the observation and to avoid the problem of false-positives, location is represented in SeqSLAM as a sequence of images, rather than a single image from one pose. Images are, beforehand, resolution-reduced and patch-normalised to enhance contrast and are, therefore, converted into visual templates. Then, to compare each query image I_q from the query sequence $\mathcal{Q} = (I_1, ..., I_Q)$ where $Q = |\mathcal{Q}|$ to each database image I_D from the database $\mathcal{D} = (I_1, ..., I_D)$ where $D = |\mathcal{D}|$, SeqSLAM calculates the difference score d

$$d = \frac{1}{R_x R_y} |I_Q - I_D| \tag{1}$$

where R_x and R_y are the horizontal and vertical image dimensions, respectively. The difference scores are assembled into the so-called difference matrix.

A next key processing step is to normalize the image difference values (d) within their (spatially) local image neighborhoods. Subsequently, a search for diagonals of low difference values is performed over the defined sequence length (Q) as depicted in Figure 1.

However, one of SeqSLAM's most significant drawbacks is its lack of viewpoint invariance inherited from the use of global image matching. To compensate for a small viewpoint invariance, variable offset image matching and distance-based template learning have been used in the SMART system built upon SeqSLAM [28]. In fact, to improve performance on traverses with small shifts in lateral pose, a variable offset image matching is used where each query frame (template) is compared

to each database frame (template) at a range of offsets with the sum of absolute differences (SAD) performed on the overlapping region and the minimum difference score is used:

$$d = \min_{\substack{x_{max\,left} \le u \le x_{max\,right} \\ y_{max\,up} \le v \le y_{max\,down}}} d(u,v) \tag{2}$$

$$d(u,v) = \frac{1}{|X_A||Y_A|} \sum_{\substack{x_a \in X_A \\ x_b \in X_B}} \sum_{\substack{y_a \in Y_A \\ y_b \in Y_B}} |A_{x_a,y_a} - B_{x_b,y_b}| \tag{3}$$

where u and v are the coordinates in the difference score, X_A, Y_A, X_B and Y_B are vectors representing the overlapping region of images a and B and x_a and x_b are the pixel coordinates in a and B respectively (Figure 2).

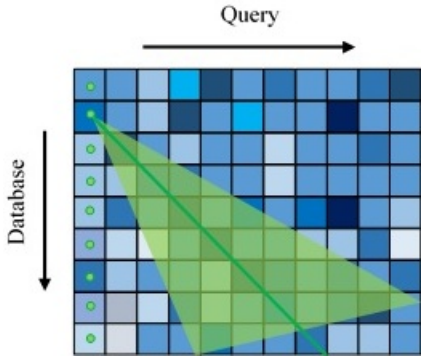


Figure 1. For each matrix entry, a sub-route score for a range of possible slopes (semi-transparent region). When all scores are calculated for one starting point, the minimal score of all of them is selected. Finally, the sub-route with the smallest score and the second smallest score are used to compute the best database matching to the input query sequence.

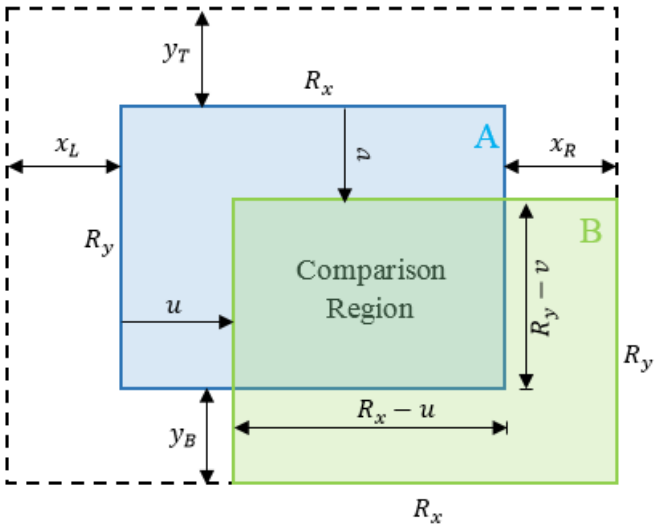


Figure 2. For each comparison, query frame B is slid over template frame a in a range of offsets (within the dashed boundary), with the minimum calculated difference score.

On the other hand, SMART learns (creates the database of templates) and queries templates at regular distance intervals (fixed distance f_{dist} between templates) rather than fixed time or frame intervals as SeqSLAM does (fixed number of frames between database and query templates).

To function effectively, odometry information (sourced either from wheel encoders or vision) is needed [28].

4. Proposed Approach

We propose a kind of hybrid approach that combines local metrical localization (e.g., visual odometry) with a topological one (e.g., SeqSLAM). The vehicle tracking is performed by VO leading to drift due to the run-time error accumulation. Loop closure recognition via SeqSLAM allows incremental pose drift [29] to be overcome and the state and position of the vehicle to be recovered in cases where the tracking is lost. The use of a two algorithms fusion-based approach is mainly motivated by the fact that VO is prone to drift and place recognition approaches generally suffer from false positive (FP) detections. Although false positives have an impact on the short-term deviation of the system inside the filtering framework, this latter robustly recovers after some time. Combined methods give, therefore, an increased robustness to the system. The template library can be loaded beforehand (when a learning has already been done) or created in real-time (without prior learning). In such a case, the template database is populated with the acquired frames from the monocular stream converted into visual templates along with their poses acquired from the VO module. The query sequence is a FIFO buffer holding the n last templates. In fact, when a new frame is captured, features are detected and matched with the last acquired frame and the relative motion is estimated using the VO module, which will subsequently allow the pose of the vehicle to be predicted. The frame is also converted into a visual template and added to the query sequence with the aim of matching with a database template. The SeqSLAM module performs the matching that will be tested to determine whether it is a true positive based on both a matching score provided by SeqSLAM and the prediction made by the VO module. If a place is recognized, a correction of the pose is made. Otherwise, the template is assumed to belong to a new place and is, therefore, added to the database as depicted in Figure 3.

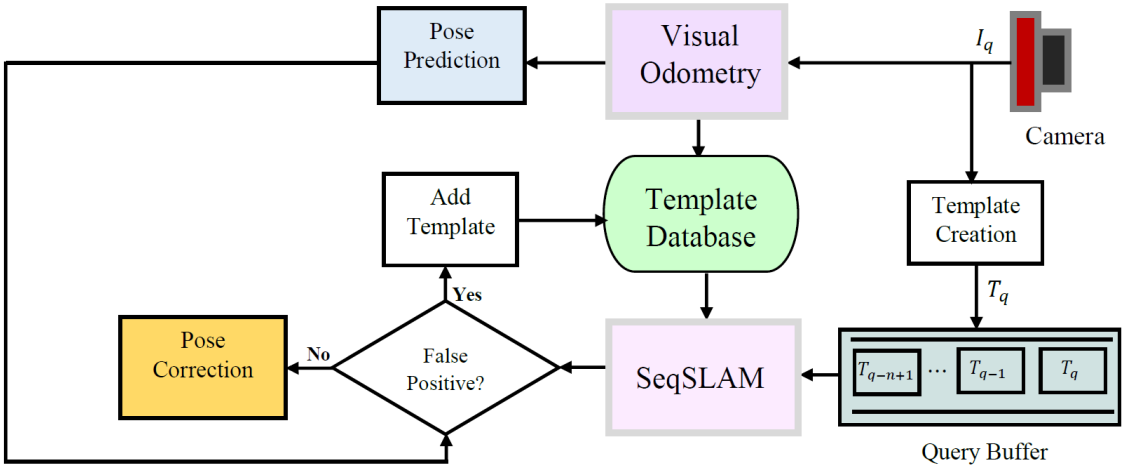


Figure 3. Block diagram of the main components of the system.

4.1. Position Tracking

The goal of this section is to track the state of the vehicle over several images. The vehicle state in time k is composed by the position in the $2D$ plane and the orientation with respect to the first reference frame. Thus, we consider the reduced state vector $x_k \in \mathbb{R}^3$

$$x_k = (p_k, \theta_k), \tag{4}$$

where $p_k \in \mathbb{R}^2$ denotes the position and $\theta_k \in \mathbb{R}$ denotes the orientation (heading angle).

We adopt a Bayesian approach [30] to track and update the position of the vehicle. We compute the posterior probability density function *(pdf)* of the state in two steps. To compute the prediction update of the Bayesian filter, we use VO. To compute the measurement update, we integrate the topological localization update, whenever it is supplied by SeqSLAM, described in the previous section.

The system model f describes the evolution of the state over time. The measurement model h relates the current measurement $z_k \in \mathbb{R}^3$ to the state. Both are expressed in a probabilistic form:

$$x_{k|k-1} = f(x_{k-1|k-1}, u_{k-1}), \tag{5}$$

$$z_k = h(x_{k|k-1}), \tag{6}$$

where $u_{k-1} \in \mathbb{R}^3$ denotes the output of the VO algorithm at time $k-1$, $x_{k|k-1}$ denotes the prediction of estimate x at time k and $x_{k-1|k-1}$ denotes the updated estimate of x at time $k-1$. The functions f and h are in general non-linear functions.

4.1.1. Visual Odometry System

Visual Odometry (VO) is usually referred to as the problem of incrementally estimating the egomotion of a vehicle using a single or multiple cameras [31]. In order to deal with all central camera models including perspective, dioptric, omnidirectional and catadioptric imaging devices, image measurements are represented as 3D bearing vectors: a unit vector originating at the camera center and pointing toward the landmark. Each bearing vector has only two degrees of freedom, which are the azimuth and elevation inside the camera reference frame as formulated in the OpenGV library [32]. Because a bearing vector has only two degrees of freedom, we refer to it as 2D information and it is normally expressed in a camera reference frame. The fundamental matrix solver within the OpenGV library computes the relative pose of a viewpoint with respect to another viewpoint given a number of eight correspondences between bearing vectors expressed in the respective camera frames within a Random Sample Consensus (RANSAC) framework to deal with false matchings [33].

4.1.2. State Prediction and Uncertainty Estimation

At time k, two consecutive images I_k and I_{k-1} are given as input to the VO algorithm. This latter returns an incremental motion estimate with respect to the local camera reference frame. We define this estimate as $\delta^*_{k,k-1} \in \mathbb{R}^3$

$$\delta^*_{k,k-1} = (\Delta s^*_k, \Delta\theta_k), \tag{7}$$

where $\Delta s^*_k \in \mathbb{R}^2$ denotes the translational component of the motion and $\Delta\theta_k$ the heading angle increment.

As we are using monocular visual odometry using only the input of a single camera, we deal with a scale ambiguity where the norm of the translational component cannot be recovered and we only obtain the direction of the translation. Δs^*_k is valid up to a scale factor and, thus, the metric translation of the vehicle in the ground-plane $\Delta s_k \in \mathbb{R}^2$ at time k with respect to the local camera frame is equal to

$$\Delta s_k = \lambda \Delta s^*_k, \tag{8}$$

where $\lambda \in \mathbb{R}$ is the scale factor. Several approaches have shown their efficacy in solving the scale factor ambiguity within the monocular scheme including the use of prior knowledge of the camera height relative to ground-plane [34].

Given that we predicted the state of the vehicle x_k using x_{k-1} and the incremental motion estimate $\delta_{k,k-1} \in \mathbb{R}^3$, the uncertainty of the pose has to be estimated as well, represented by a 3×3 covariance matrix.

In order to estimate the covariance matrix $\Sigma_{\delta_{k,k-1}} \in \mathbb{R}^{3\times3}$, we use the Monte Carlo technique [35]. In general, when dealing with a nonlinear function $f(.)$ that relates a random variable Y to an N-dimensional random variable X such as $Y = f(X)$, where X has a mean $\bar{X}$ and a covariance Σ_X, the transformed mean and covariance of Y can be obtained via a Monte Carlo simulation that relies on

a repeated random sampling. In fact, a large number of samples $\{X_1, X_2 ..., X_n\}$ are randomly drawn from X and the function $f(.)$ is evaluated for each sample. For the transferred samples from function $f(.)$, $\{Y_1, Y_2 ..., Y_n\}$, the mean $\bar{Y}$ and covariance Σ_Y are estimated according to:

$$\bar{Y} = \frac{1}{N}\sum_{i=1}^{N} Y_i \tag{9}$$

$$\Sigma_y = \frac{1}{N-1}\sum_{i=1}^{N}(Y_i - \bar{Y})(Y_i - \bar{Y})^T. \tag{10}$$

In the case of VO, the algorithm uses at every step a set of 2D corresponding bearing vectors between images k and $k-1$ and provides an incremental estimate $\delta_{k,k-1}$. In fact, features which are detected and matched between every consecutive pair of frames are, subsequently, converted into bearing vectors. We randomly sample eight correspondences from corresponding bearing vectors and feed them to the RANSAC procedure to compute an estimate $\{\delta_i\}$. All the estimates for which the number of inliers exceeds a prefixed threshold t are saved in a set $S = \{\delta_i\}$. Estimating the covariance matrix using the Monte Carlo technique requires a large number of samples to be randomly drawn from the initial N-dimensional random variable x. We usually obtain a high number of valid Monte Carlo estimates (e.g., more than 500 out of the 1000 iterations result in a valid estimation) and the covariance $\Sigma_{\delta_{k,k-1}}$ is, finally, estimated using (9) and (10).

The error of the VO is propagated throughout consecutive camera positions as follows. At time k, the state $x_{k|k-1}$ depends on $x_{k-1|k-1}$ and $\delta_{k,k-1}$

$$x_{k|k-1} = f(x_{k-1|k-1}, \delta_{k,k-1}). \tag{11}$$

We compute the associated covariance $\Sigma_{x_{k|k-1}} \in \mathbb{R}^{3\times3}$ by the error propagation law:

$$\Sigma_{x_{k|k-1}} = \nabla f_{x_{k-1|k-1}} \Sigma_{x_{k-1|k-1}} \nabla f_{x_{k-1|k-1}}^T + \nabla f_{\delta_{k,k-1}} \Sigma_{\delta_{k,k-1}} \nabla f_{\delta_{k,k-1}}^T \tag{12}$$

assuming that $x_{k-1|k-1}$ and $\delta_{k,k-1}$ are uncorrelated.

We compute the Jacobian matrices numerically, the rows of the jacobian matrices $\nabla({}^i f_{x_{k-1|k-1}})$, $\nabla({}^i f_{\delta_{k,k-1}}) \in \mathbb{R}^{1\times3}$ $(i = 1, 2, 3)$ are computed as:

$$\nabla({}^i f_{x_{k-1|k-1}}) = \Big[\frac{\partial({}^i f)}{\partial({}^1 x_{k-1|k-1})} \; \frac{\partial({}^i f)}{\partial({}^2 x_{k-1|k-1})} \; \frac{\partial({}^i f)}{\partial({}^3 x_{k-1|k-1})}\Big], \tag{13}$$

$$\nabla({}^i f_{\delta_{k|k-1}}) = \Big[\frac{\partial({}^i f)}{\partial({}^1 \delta_{k,k-1})} \; \frac{\partial({}^i f)}{\partial({}^2 \delta_{k,k-1})} \; \frac{\partial({}^i f)}{\partial({}^3 \delta_{k,k-1})}\Big], \tag{14}$$

where ${}^i x_{k-1|k-1}$ and ${}^i \delta_{k,k-1}$ denote the *i-th* component of $x_{k-1|k-1}$ and $\delta_{k,k-1}$ respectively.

4.2. Loop Closure-Based Measurement Update

In the SeqSLAM algorithm, the input is the query sequence that refers to the last acquired images converted to visual templates. The algorithm selects the path in the difference matrix that connects the query templates to the database ones. The observations are the sequence of n query images I_q, and the state space S is the set of database images I_d. For each observation there is a state corresponding to one of the database images in the state space as depicted in Figure 4 where primitive high-resolution images are used instead of low-resolution templates for the purpose of clarity. Thus, the observables here are the query and database templates, which are continuously calculated from the monocular sequence and the tracked state $x_k = (x, y, \theta)$ is hidden. However, to each database template is connected a pose where this image has been taken and to the query template to be matched is connected the predicted

state. Hence, the unobservable state variables are directly obtained from the observable templates and the transformation from the state space to the observation space is a simple linear model.

$$y = Hx. \tag{15}$$

where $H = I_3$. In fact, when a place is recognized, the corresponding information of the node (matched in the database) can be used as an observation of the filter.

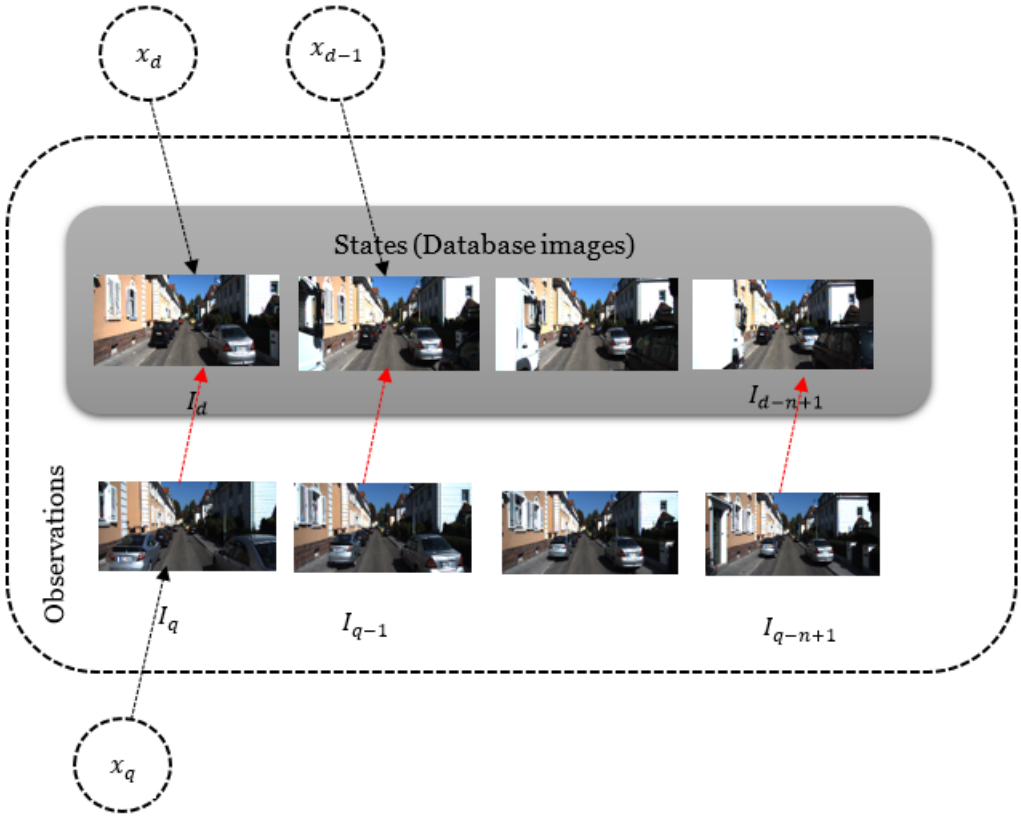


Figure 4. Unobservable state variables for a measurement update obtained from observable query and database templates by a linear model.

4.3. False Positives Filtering

The Gaussian assumption underlying the Kalman Filter and its variants implies that when an observation that significantly differs from the state estimate, as determined from the state covariances, is incorporated into the state estimate, the resulting Gaussian deviates significantly from its prior shape. This means that the current estimate of the true state is no longer useful which has a negative impact on the system robustness. Although the filter recovers after some time, several applications depend on the short-term result in which false positives (FP) have a disastrous impact. In fact, even with the use of a sequence matching instead of a single image by the SeqSLAM algorithm, this latter still suffers from false-positives. Thus, to filter them out, we use the state and covariance estimates and the Gaussian basis of filtering techniques to estimate the likelihood of a given observation. The conditional *pdf* $P(z_k|x_k, \Sigma_{x_k})$ is evaluated for the given observation and state estimates to reject observations with too low likelihood as false-positives. The conditional *(pdf)* is simply a multi-dimensional Gaussian with the mean the estimated state and its covariance. Both the mean and the covariance have to be transformed into the observation space. Hence, the likelihood has the following form

$$P(z_k|x_k, \Sigma_{x_k}) = \frac{1}{\sqrt{(2\pi)^n C_k}} e^{-1/2(z-Hx_k)^T C_k^{-1}(z-Hx_k)} \tag{16}$$

where $C_k = H\Sigma_{x_k}H^T + R$ is the covariance matrix in the observation space, n the state dimension and R is the measurement covariance matrix. If the evaluated likelihood is above a preset, empirically determined threshold, the measurement update step is performed. Otherwise, only the prediction is done and the covariance of the state estimate will grow until a successful observation is reported. As well as the case of rejected observations, the dynamics update is still performed in the case of false negatives and they, therefore, do not affect the system performance.

SeqSLAM uses a threshold on similarity between sequences to determine whether a match refers to a true positive determined by precision-recall tests. The lower the threshold is, the more similar are the matched images. However, a too low threshold leads to false negatives (e.g., non-detections). The better compromise has been achieved when using both the similarity score and the likelihood to assess true positives without too many non detections as presented in Algorithm 1.

Algorithm 1: False positives filtering.

Similarity Score$\leftarrow S_{score}$;
Similarity threshold low$\leftarrow T_{low}$;
Similarity threshold high$\leftarrow T_{high}$;
Likelihood threshold$\leftarrow T_{likelihood}$;
if *($(S_{score} < T_{low})$ or $((S_{score} < T_{high})$ and $(likelihood < T_{likelihood}))$)* **then**
  Do correction;
end

4.4. Filtering-Based Fusion of Visual Odometry and SeqSLAM

Our aim is to reduce the uncertainty associated to the state estimate by fusing the prediction estimate with the measurement, whenever a valid measurement issued from a recognized loop closure by SeqSLAM is available. The outputs of this fusion step are the updated estimate $x_{k|k}$ and its covariance $\Sigma_{x_{k|k}} \in \mathbb{R}^3$. We compute them according to the Kalman filter (KF) update mode as we are dealing with a linear measurement model. The update step is first performed by predicting the measurement and its uncertainty given by

$$z_{k|k-1} = Hx_{k|k-1}, \tag{17}$$

$$S_k = H\Sigma_{k|k-1}H^T + R. \tag{18}$$

Then, the measurement update adjusts the prediction results according to

$$x_{k|k} = x_{k|k-1} + K_k(z_k - z_{k|k-1}) \tag{19}$$

$$= (I - K_kH)x_{k|k-1} + K_kz_k. \tag{20}$$

$$\Sigma_{k|k} = (I - K_kH)\Sigma_{k|k-1}(I - K_kH)^T + K_kRK_k^T \tag{21}$$

with K_k a gain matrix that is optimal in the minimum variance sense and is given by

$$K_k = \Sigma_{k|k-1}H^TS_k^{-1} = M_kS_k^{-1}, \tag{22}$$

where M_k is the cross-covariance between the state and output predictions.

4.5. Unscented Kalman Filter

The Unscented Kalman Filter (UKF) is an alternative to the Extended Kalman Filter (EKF) demonstrating a superior performance and ease of implementation for nonlinear estimation. The UKF

uses an alternative form for dealing with nonlinearity based on the sigma points. The time update includes the weights and sigma points calculations, whereas the measurement update uses the sigma points to generate the covariance matrices and the Kalman gain respectively. The augmented state matrix Y is constructed from the prior belief of the state $(x_{k-1|k-1}, \Sigma_{k-1|k-1})$:

$$\mathrm{Y} = \begin{bmatrix} \Sigma_{k-1|k-1} & 0 \\ 0 & Q_k \end{bmatrix} \tag{23}$$

where Q_k is the process noise covariance matrix estimated via a Monte Carlo simulation as explained in the previous section. The prior belief $(x_{k-1|k-1}, \Sigma_{k-1|k-1})$ is converted into a sigma point representation via:

$$\mathrm{AA}^T = Y, \qquad \text{(cholesky decomposition)} \tag{24}$$

$$\chi_0 = x_{k-1|k-1} \tag{25}$$

$$\chi_{i,k|k-1} = x_{k-1|k-1} + \sqrt{(n+\kappa)}\mathrm{col}_i\mathrm{A}, \qquad i = 1, ..., n \tag{26}$$

$$\chi_{i,k|k-1} = x_{k-1|k-1} - \sqrt{(n+\kappa)}\mathrm{col}_i\mathrm{A}, \qquad i = n, ..., 2n \tag{27}$$

where κ is a scaling parameter that affects fourth and higher order moments of the *pdf*. The associated weights β_i are the following:

$$\beta_i = \begin{cases} \frac{\kappa}{n+\kappa} & i = 0 \\ \frac{1}{2}\frac{\kappa}{n+\kappa} & \text{otherwise.} \end{cases} \tag{28}$$

The subsequent step consists in passing the sigma point through the state evolution model

$$\gamma_{i,k|k-1} = \mathbf{f}(\chi_{i,k|k-1}) \qquad i = 0, ..., 2n. \tag{29}$$

The mean and covariance estimates for γ are calculated as

$$x_{k|k-1} = \sum_{i=0}^{2n} \beta_i \gamma_{i,k|k-1}, \tag{30}$$

$$\mathbf{\Sigma}_{k|k-1} = \sum_{i=0}^{2n} \beta_i (\gamma_{i,k|k-1} - x_{k|k-1})(\gamma_{i,k|k-1} - x_{k|k})^T. \tag{31}$$

The estimate of the posterior $(x_{k|k}, \Sigma_{k|k})$ is performed through passing each sigma point according to the observation model

$$v_{i,k|k} = H.\chi_{i,k|k-1}, \qquad i = 0, ..., 2n. \tag{32}$$

The predicted measurement and innovation covariance are then computed

$$y_{k|k-1} = \sum_{i=0}^{2n} \beta_i v_{i,k|k}, \tag{33}$$

$$\mathbf{V}_{k|k} = \sum_{i=0}^{2n} \beta_i (v_{i,k|k} - x_{k|k-1})(v_{i,k|k-1} - x_{k|k})^T + \mathbf{R}_k, \tag{34}$$

where $\mathbf{R}_k$ is the measurement noise covariance matrix. The state measurement covariance and Kalman gain are, next, built according to:

$$\mathbf{U}_{k|k} = \sum_{i=0}^{2n} \beta_i (\chi_{i,k|k-1} - x_{k|k-1})(\gamma_{i,k|k} - y_{k|k-1})^T, \tag{35}$$

$$\mathbf{K}_{k|k} = \mathbf{U}_{k|k}\mathbf{V}_{k|k}^{-1}. \tag{36}$$

Finally, the posterior belief, $x_{k|k}$, $\mathbf{\Sigma}_{k|k}$ is computed according to:

$$x_k = x_{k|k-1} + \mathbf{K}_k(y_k - y_{k|k-1}) \tag{37}$$

$$\mathbf{\Sigma}_k = \mathbf{\Sigma}_{k|k-1} - \mathbf{K}_k\mathbf{U}_k^T. \tag{38}$$

5. Experiments and Results

5.1. The Experimental Dataset and Parameters

5.1.1. The Experimental Dataset

The Kitti odometry dataset [15], recorded from cars driven in urban and rural areas and on highways, consists of 22 sequences where the first 11 are provided with ground truth.

In order to evaluate our method, we conducted experiments on three data sequences from the odometry category with a loopy trajectory. Together, these sequences, presented in Figure 5, include 1249 images of loop closures. The total driving distance for loop closures in these sequences is 1169.6 m and are highlighted in blue in Figure 5. The locations of loop closures for the Sequence 00 in terms of the amount of travelled meters from the first reference pose are presented in Table 1.

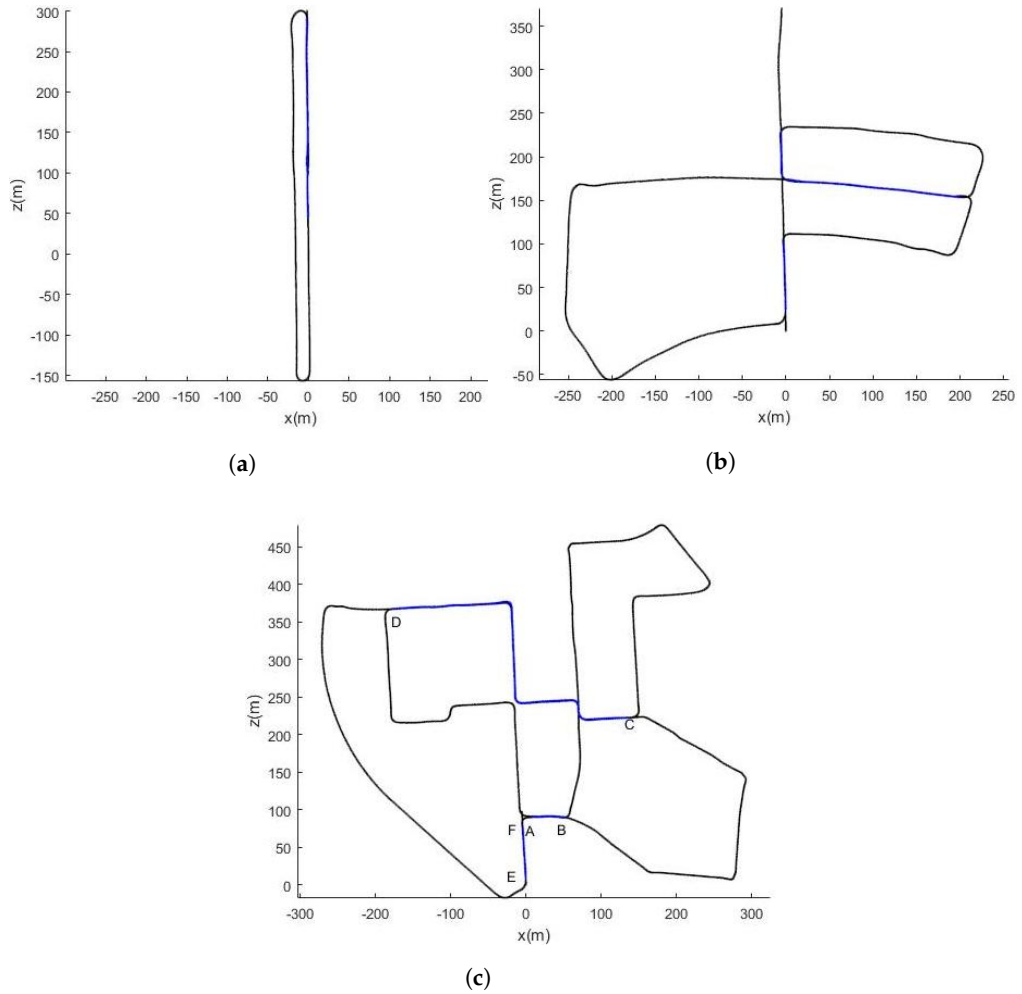


Figure 5. Ground truth trajectories of sequences 00, 05 and 06 of the KITTI dataset comprising loop-closures. (**a**) Sequence 06; (**b**) Sequence 05; (**c**) Sequence 00.

Table 1. Loop closure locations in meters for Sequence 00.

Sequence	Distance up to	Traveled Distance (m)
00	A	1160.1
	B	1204.8
	C	2566.1
	D	3023.3
	E	3636.3
	F	3707

We recall that the Kitti dataset provides challenging benchmarks to the computer vision community that are available online www.cvlibs.net/datasets/kitti.

5.1.2. Parameters

The parameters used for SeqSLAM are those reported in the literature and presented in Table 2.

Table 2. SeqSLAM's parameters.

Parameter	Value	Description
R_x,R_y	64, 16	Template size
Q	10	Query sequence length
P	8×8 pixels	Patch normalization size
X_L, X_R, Y_T, Y_R	1, 1, 1, 1	Shift offsets

In order to calibrate SeqSLAM and evaluate the uncertainty of the position determination, we used almost 50% of the 1249 images providing two traversals of the same route. The first traversal was used as a database sequence and the second as a query sequence. We assimilated the measurement noise $\nu_k = \mathcal{N}(0, R)$. The matching results were compared to the ground truth. In the case of true positives, the deviations have been chosen to be respectively $\sigma_x = 1$, $\sigma_y = 1$ and $\sigma_\theta = 0.1$ so that the returned result is always within 3 standard deviations from the ground truth.

5.2. Results

5.2.1. Accuracy Evaluation

We display the ground truth trajectory measured by a precise GPS in green in Figures 6–8 for Sequences 00, 05 and 06 respectively from the Kitti odometry benchmark. We show the path estimated by VO in red and the corrected one (VO+SeqSLAM) in blue. Even though the position is generally accurately detected, the non-detections (FN) are sometimes visible (the correction is not made in the beginning of a loop-closure). In fact, when an update from SeqSLAM is integrated into the Bayesian tracking to correct the trajectory after an important drift, the correction is suddenly made, resulting in a non-smooth trajectory (marked by the blue stars). The drift, though, is greatly reduced for a precise localization afterwards.

We show the mean error per travelled meter of position and heading in Figure 9 for the Sequence 00. For a travelled trajectory of 1100 m, no loop closures are detected and the mean error in position is 5 (m) and 4 (deg) in heading. For 3000 (m) of travelled distance where two loop closures are detected (Segment AB and Seqment CD in Figure 5c), the mean error is 5 (m) in position and 3.6 (deg) in heading when the SeqSLAM-based correction is integrated into the Bayesian tracking against 9 (m) in position and 4.5 (deg) for VO only. When the vehicle performs a big rotation movement (in segment DE of Figure 5c), the mean error increases dramatically to reach 11 (m) for VO and 6.7 (m) for the SeqSLAM-based corrected VO for a travelled distance of almost 4000 (m). The problem of important drift caused by big rotations is inherent to the monocular VO.

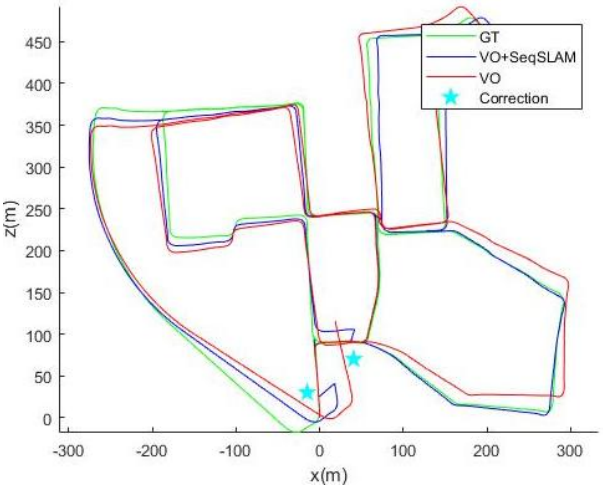


Figure 6. Ground truth, Visual Odometry (VO) and (VO+seqSLAM) for Sequence 00.

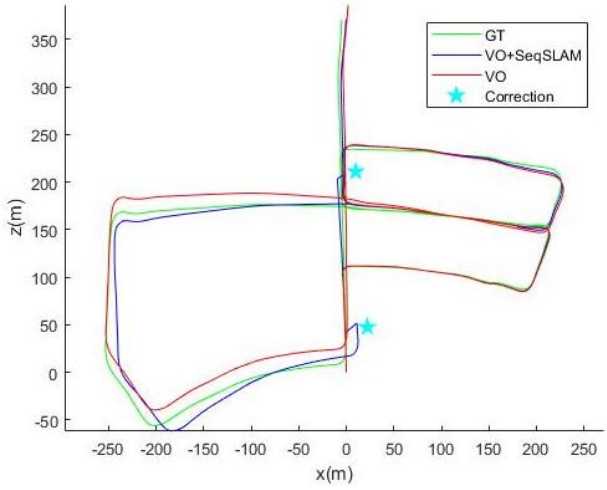


Figure 7. Ground truth, VO and (VO+seqSLAM) for Sequence 05.

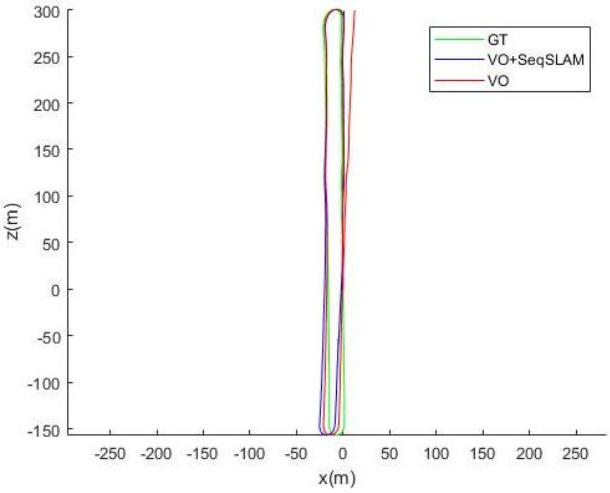
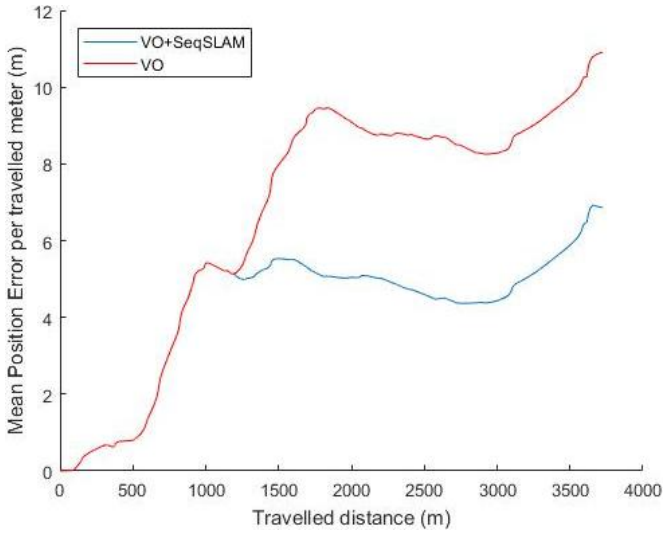
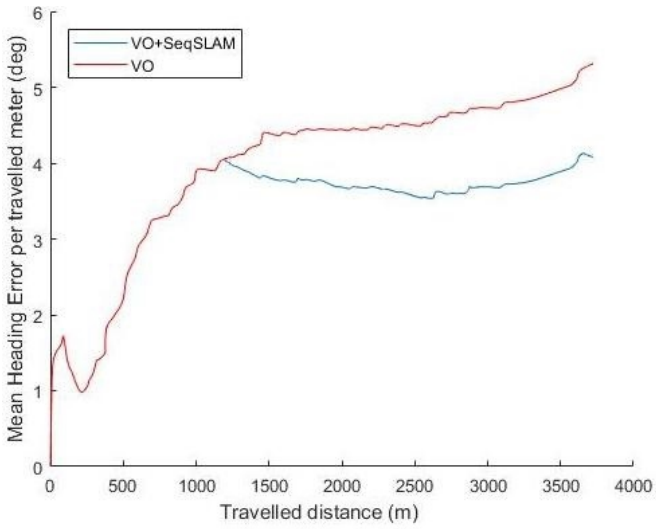


Figure 8. Ground truth, VO and (VO+seqSLAM) for Sequence 06.



(**a**)



(**b**)

Figure 9. Mean error per travelled meter with respect to the travelled distance for Seq00. (**a**) mean position error per travelled meter; (**b**) mean heading error per travelled meter.

We show the comparative results between the corrected and non-corrected journeys in Table 3 in terms of average error in meters and the percentage of the error with respect to the whole trajectory. The performance of SeqSLAM-based loop closure detection and correction is almost two times better than only VO. We also present in Table 4 the error of the ending pose for all the test sequences.

Table 3. Average position errors with and without correction.

Sequence	Method	Mean Position Error (m)	(%) of Trajectory
00	VO+SeqSLAM	7.98	0.2
	VO	14.26	0.39
05	VO+SeqSLAM	5.59	0.25
	VO	9.02	0.41
06	VO+SeqSLAM	3.43	0.27
	VO	6.54	0.53

Table 4. Error of the ending pose with and without correction.

Sequence	Method	Position Error (m)	Heading Error (deg)
00	VO+SeqSLAM	4.5	2.5
	VO	16.71	9.96
05	VO+SeqSLAM	1.37	2.85
	VO	14.29	6.03
06	VO+SeqSLAM	3.93	1.31
	VO	18.03	3.55

5.2.2. Fusion Method Evaluation

The UKF is a superior alternative to the EKF for a variety of estimation and control problems. However, its effectiveness depends on the nonlinearity of the problem. In our case, we are dealing with a nonlinear prediction model and a linear measurement model. As listed in Table 5, our experimental results and analysis indicate that UKF performs as efficiently as the EKF. However, the additional computational overhead of the UKF and the linear nature of the update step with SeqSLAM's observations lead to the conclusion that the EKF is a better choice for fusing VO and SeqsLAM.

Table 5. EKF-based fusion vs UKF-based.

Sequence	Fusion Method	Mean Position Error (m)	(%) of Trajectory
00	EKF	7.98	0.2
	UKF	7.96	0.2
05	EKF	5.59	0.25
	UKF	5.62	0.25

5.3. Timing and Storage

In this section, we briefly describe the storage and computational requirements of the system. In fact, all the templates are down-sampled from 1241×376 to 64×16 (0.22%) pixels. Consequently, the storage and computational requirements are greatly reduced. To deal with the computational complexity of building a difference matrix between the query and all the database images, a CUDA-based solution was designed in [36], allowing less than 30 ms to be achieved for the query and database sequences used in our experiments. This timing is based on the use of a mid-range GPU, the CUDA NVIDIA GeForce GTX 850M running at 876 MHz with 4096 MB of GDDR device memory. The acceleration is mainly based on the allocation of the three major steps of SeqSLAM, namely: the difference matrix computation, the difference matrix contrast-enhancement and the route searching to three GPU kernels that exploit the parallel CUDA threads and thread-blocks and the fast shared memory. The database, in this design, is stored in the GPU's global memory. Regarding the timing using a commercially available laptop with an 8 core-2.40 GHz clock, it is around 100 ms.

5.4. Discussion

The template library is created online using navigation information from the VO. That is, when there is not a match in the database, a node is added to this latter with the template and the corresponding information of pose obtained from the VO module. The sequence searching strategy used in this work is based on a regular time interval but could be optimized for a regular distance interval between templates to make the localized sequence searching algorithm more efficient as reported in [28]. In fact, we have not relied on the regular distance strategy as we have used the Kitti dataset for evaluation. a regular distance consists in taking templates at a regular distance of

1 m for example. Moreover, this work is a proof of concept that an enhancement could be obtained by only detecting loop closures without jointly optimizing over landmarks and poses as a SLAM system does. However, the greatest interest would be generated when a learning of a geotagged database is already performed which would allow an absolute localization in the global reference frame. The SeqSLAM algorithm could also be adapted to work with different camera systems to make it possible to share the database of templates between different users. Furthermore, even though the drift has been greatly reduced, other methods could be fused with SeqSLAM and VO to make the output suitable for autonomous cars.

6. Conclusions

This paper presented a solution to localize a vehicle in urban environments by means of the integration of VO and SeqSLAM. SeqSLAM is a well-known successful approach for place recognition in varying conditions such as seasonal changes and day and night cycles when not dealing with a viewpoint change. The localization was performed using a single on-board camera and the template library was created online using navigation information from the VO. The integration was performed with a Bayesian filtering through the use of an EKF. That is, when a place is recognized, the corresponding information is used as an observation of the filter, otherwise the prediction is performed by the VO module. The performance of the system can be increased by using a pre-learnt geo-tagged database of images to develop a reliable alternative to satellite-based global positioning. In such a case, the importance of SeqSLAM is even more important due to its ability to deal with severe lighting differences between the learnt database and the actual visual input. The results showed the superiority of the visual odometry integrated with SeqSLAM algorithm in real-time localization as the navigation errors are greatly reduced by loop-closure detection.

Author Contributions: S.O. conceived, designed and developed the proposed method and its experimental validation; All other authors contributed their expertise to validate the proposal, evaluate and question the experimental results. All authors contributed in framing the writing of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GPS	Global Positioning System
SfM	Structure from Motion
VO	Visual Odometry
SeqSLAM	Sequence SLAM
SLAM	Simultaneous Localization and Mapping
KF	Kalman Filter
UKF	Unscented Kalman Filter
FP	False positive
FN	False negative
pdf	probability density function

References

1. Milford, M.J.; Wyeth, G.F. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), St. Paul, MN, USA, 14–18 May 2012; pp. 1643–1649.
2. Milford, M. Vision-Based Place Recognition: How Low Can You Go? *Int. J. Robot. Res.* **2013**, *32*, 766–789.
3. Bonardi, F.; Ainouz, S.; Boutteau, R.; Dupuis, Y.; Savatier, X.; Vasseur, P. PHROG: a multimodal Feature for Place Recognition. *Sensors* **2017**, *17*, 1167, doi:10.3390/s17051167.

4. Stone, T.; Mangan, M.; Ardin, P.; Webb, B. Sky segmentation with ultraviolet images can be used for navigation. In Proceedings of the Robotics: Science and Systems X (RSS), University of California, Berkeley, CA, USA, 12–16 July 2014.
5. Sünderhauf, N.; Shirazi, S.; Dayoub, F.; Upcroft, B.; Milford, M. On the performance of convnet features for place recognition. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 4297–4304.
6. Sünderhauf, N.; Neubert, P.; Protzel, P. Are We There Yet? Challenging SeqSLAM on a 3000 km Journey Across All Four Seasons. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Workshop on Long-Term Autonomy, Chemnitz, Germany, 15 January 2013.
7. Kalman, R.E. A new approach to linear filtering and prediction problems. *J. Basic Eng.* **1960**, *82*, 35–45.
8. Cummins, M.; Newman, P. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *Int. J. Robot. Res.* **2008**, *27*, 647–665.
9. Cummins, M.; Newman, P. Appearance-only SLAM at large scale with FAB-MAP 2.0. *Int. J. Robot. Res.* **2011**, *30*, 1100–1123.
10. Badino, H.; Huber, D.; Kanade, T. Visual topometric localization. In Proceedings of the IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany, 5–9 June 2011; pp. 794–799.
11. Baatz, G.; Köser, K.; Chen, D.; Grzeszczuk, R.; Pollefeys, M. Leveraging 3D City Models for Rotation Invariant Place-of-Interest Recognition. *Int. J. Comput. Vis.* **2012**, *27*, 315–334, .
12. Oh, S.; Tariq, S.; Walker, B.; Dellaert, F. Map-Based Priors for Localization. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September–2 October 2004.
13. Milford, M. Visual Route Recognition with a Handful of Bits. In Proceedings of Robotics: Science and Systems (RSS), Sydney, Australia, 9–13 July 2012; MIT Press: Cambridge, MA, USA, 2012.
14. Clipp, B.; Lim, J.; Frahm, J.-M.; Pollefeys, M. Parallel, real-time visual SLAM. In Proceedings of the IEEE/RSJ International Conference on IROS, Taipei, Taiwan, 18–22 October 2010; pp. 3961–3968.
15. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
16. Geiger, A.; Ziegler, J.; Stiller, C. Stereoscan: Dense 3D reconstruction in real-time. In Proceedings of the IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany, 5–9 June 2011; pp. 963–968.
17. Kaess, M.; Ni, K.; Dellaert, F. Flow separation for fast and robust stereo odometry. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, 12–17 May 2009.
18. Newcombe, R.A.; Lovegrove, S.J.; Davison, A.J. Dtam: Dense tracking andmapping in real-time. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2320–2327.
19. Ranganathan, A.; Dellaert, F. Online probabilistic topological mapping. *IJRR* **2011**, *30*, 755–771.
20. Paul, R.; Newman, P. FAB-MAP 3D: Topological mapping with spatial and visual appearance. In Proceedings of the IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 2649–2656.
21. Kaess, M.; Johannsson, H.; Roberts, R.; Ila, V.; Leonard, J.J.; Dellaert, F. iSAM2: Incremental smoothing and mapping using the bayes tree. *Int. J. Robot. Res.* **2012**, *31*, 217–236.
22. Mei, C.; Sibley, G.; Cummins, M.; Newman, P.; Reid, I. RSLAM: A System for Large-Scale Mapping in Constant-Time Using Stereo. *Int. J. Comput. Vis.* **2011**, *94*, 198–214.
23. Brubaker, M.A.; Geiger, A.; Urtasun, R. Map-Based Probabilistic Visual Self-Localization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 652–665.
24. Floros, B.L.G.; van der Zander, B. Openstreetslam: Global vehicle localization using openstreetmaps. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 1054–1059.
25. Hentschel, M.; Wagner, B. Autonomous robot navigation based on OpenStreetMap geodata. In Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems, Funchal, Portugal, 19–22 September 2010; pp. 1645–1650.
26. Chu, H.; Mei, H.; Bansal, M.; Walter, M.R. Accurate vision-based vehicle localization using satellite imagery. *arXiv* **2015**, arXiv:1510.09171.

27. Majdik, A.L.; Verda, D.; Albers-Schoenberg, Y.; Scaramuzza, D. Micro air vehicle localization and position tracking from textured 3d cadastral models. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 920–927.
28. Pepperell, E.; Corke, P.; Milford, M. Towards Persistent Visual Navigation Using SMART. In Proceedings of the Australasian Conference on Robotics and Automation (ARAA), Sydney, Australia, 2–4 December 2013.
29. Kümmerle, R.; Grisetti, G.; Strasdat, H.; Konolige, K.; Burgard, W. G2o: A general framework for graph optimization. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011.
30. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, MA, USA, 2005; Volume 1.
31. Scaramuzza, D.; Fraundorfer, F. Visual odometry (tutorial). *IEEE Robot. Autom. Mag.* **2011**, *18*, 80–92.
32. Kneip, L.; Furgale, P. OpenGV: a unified and generalized approach to real-time calibrated geometric vision. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 1–8.
33. Fischler, M.A.; Bolles, R.C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395.
34. Dingfu, Z.; Dai, Y.; Hongdong, L. Reliable scale estimation and correction for monocular visual odometry. In Proceedings of the IEEE Intelligent Vehicles Symposium, Gothenburg, Sweden, 19–22 June 2016; pp. 490–495.
35. Thrun, S.; Fox, D.; Burgard, W.; Dellaert, F. Robust Monte Carlo Localization for Mobile Robots. *Artif. Intell.* **2001**, *128*, 99–141.
36. Ouerghi, S.; Boutteau, R.; Tlili, F.; Savatier, X. CUDA-based SeqSLAM for Real-Time Place Recognition. In Proceedings of the 25th International Conference on Computer Graphics, Visualization and Vision (WSCG), Plzen, Czech Republic, 29 May–2 June 2017.