



Stereo vision-based vehicle localization in point cloud maps using multiswarm particle swarm optimization

V. John¹ · Z. Liu² · S. Mita¹ · Y. Xu¹

Received: 19 June 2018 / Revised: 12 November 2018 / Accepted: 3 January 2019 / Published online: 10 January 2019
© Springer-Verlag London Ltd., part of Springer Nature 2019

Abstract

We propose a vision-based localization algorithm with multiswarm particle swarm optimization for driving an autonomous vehicle. With stereo vision, the vehicle can be localized within a 3D point cloud map using the particle swarm optimization. For vehicle localization, the GPS (global positioning system)-based algorithms are often affected by the certain conditions resulting in intermittent missing signal. We address this issue in vehicle localization by using stereo vision in addition to the GPS information. The depth-based localization is formulated as an optimization-based tracking problem. Virtual depth images generated from the point cloud are matched with the online stereo depth images using the particle swarm optimization. The virtual depth images are generated from the point cloud using a series of coordinate transforms. We propose a novel computationally efficient tracker, i.e., a multiswarm particle swarm optimization-based algorithm. The tracker is initialized with GPS information and employs a Kalman filter in the bootstrapping phase. The Kalman filter stabilizes the GPS information in this phase and, subsequently, initializes the online tracker. The proposed localization algorithm is validated with acquired datasets from driving tests. A detailed comparative and parametric analysis is conducted in the experiments. The experimental results demonstrate the effectiveness and robustness of the proposed algorithm for vehicle localization, which advances the state of the art for autonomous driving.

Keywords Vehicle Localization · Intelligent vehicles · Autonomous Vehicles · Particle Swarm Optimization · Localization

1 Introduction

In recent years, the development of autonomous vehicles and advanced driver assistance systems (ADAS) [9,12,16,17,21] has received significant attention from the research community. An important research in ADAS is the localization of the vehicle in a dynamic environment. Vehicle localization is important for several applications such as path planning, environment perceptions, and automated driving. For environment perception, details from the surrounding

environment can be used by the vehicle in aiding the task such as lane marker detection and curb detection. We have investigated the problem of localizing the vehicle in a dense 3D point cloud, which is generated by a mobile mapping system and contains detailed environment information. Currently, researchers adopt the GPS sensor to localize the vehicle [26]. However, the GPS sensor is susceptible to intermittent missing signals in urban areas, the degradation of GPS signal and missing satellite coverage, resulting in localization errors [5,11].

We propose a new method to address this issue by using the depth information from stereo vision system to perform the localization. Virtual depth maps generated from the 3D point cloud are matched with the stereo depth maps to localize the vehicle. Virtual depth maps are generated using the following set of coordinates transformation matrices, the fixed vehicle stereo and time-varying world-vehicle transformation matrices. It can be seen that the optimal set of coordinate transformation matrices generate a virtual depth map which is identical to the online stereo depth information. Thus, by estimating the optimal set of coordinate transforms, we perform

✉ V. John
vijayjohn@toyota-ti.ac.jp

Z. Liu
zheng.liu@ieee.org

S. Mita
smita@toyota-ti.ac.jp

Y. Xu
yuquanxu@toyota-ti.ac.jp

¹ Toyota Technological Institute, Nagoya, Japan

² University of British Columbia, Vancouver, Canada

the vehicle localization. We propose to perform the vehicle localization in three phases: the off-line phase, the bootstrapping phase, and the online phase. In the off-line phase, we use the particle swarm optimization algorithm (PSO) to estimate the vehicle stereo transformation matrix, while in the bootstrapping phase and online phases, we estimate the world-vehicle transformation matrix. The Kalman filter is used in the bootstrapping phase to stabilize the GPS–INS information and subsequently initialize a novel multiswarm PSO-based tracking algorithm in the online phase. The time-varying world-vehicle transformation matrix is estimated by the PSO-based tracker in the online phase. The particle swarm optimization (PSO) algorithm generates candidate virtual depth maps, which are then evaluated within a depth-based cost function. The depth-based cost function measures the similarity between the virtual depth map and the online stereo depth map, which is then utilized by the PSO framework to estimate the optimal set of coordinate transforms. To estimate the optimal transforms in a computationally efficient manner, we prune the depth maps and restrict the cost function evaluation to the remaining depth regions. We perform the pruning using the v -disparity method [18] and eliminate certain objects such as the pedestrians, vehicles, and buildings.

We evaluated our proposed algorithm with acquired data sets and performed a detailed comparative and parameter analysis. Experimental results show that the proposed algorithm performs better than the baseline algorithms. We extend the work by John et al. [15], where authors utilized the standard PSO to localize the vehicle. Our main contributions include: (1) A novel multiswarm PSO algorithm is used to localize the vehicle in the point cloud; (2) the Kalman filter is used in the bootstrapping phase to stabilize the GPS–INS information and initialize the online PSO tracker. Additionally, we perform a detailed parametric and comparative analysis in comparison with previous work [15].

The rest of this article is structured as follows. In Sect. 2, the state-of-the-art literature is reviewed. We then present our proposed algorithm in Sect. 3. Experimental results are presented in Sect. 4, and our observations and conclusion are presented in Sects. 5 and 6, respectively.

2 Literature review: state of the art

Usually, a GPS sensor is employed for vehicle localization. However, the GPS-based system is prone to low accuracy and limited by intermittent weak signals due to insufficient or missing satellite, tall buildings, and tunnels [5,11]. Researchers have sought to address this issue by incorporating additional sensors [3]. Typically, the inertial navigational systems (INS) are used to enhance the vehicle's capability in localization. The INS systems filter the GPS signals to

account for the localization signals. Additionally, the INS also performs interpolation to account for the intermittent missing GPS signals. The filtering and interpolation are mainly performed using the Kalman filter [4]. However, the Kalman filter is limited to linear motion and fails to perform accurate localization for nonlinear vehicle motion. This is addressed by nonlinear variants of the Kalman filter such as the Extended Kalman filter (EKF) [24]. In case of the visual odometry methods, following the initial localization, feature extraction and matching in successive frames are used for localization [22]. While the INS and VO-based methods do improve the localization accuracy, they are prone to accumulative drift errors. To address this issue, sensors such as LIDAR are used [26]. The LIDAR sensor can localize the vehicle in the surrounding environment using features from surrounding environments, such as lane markings and buildings. However, precise LIDAR sensors tend to be too expensive to afford [26]. Another approach considers the localization as a Simultaneous Localization and Mapping (SLAM) problem. Despite of the recent developments in vehicle localization using SLAM, the approach is still prone to the effects of error accumulation or divergence [2].

The problem of divergence is addressed by utilizing prior information such as dense point clouds [6,19,23]. To integrate the prior information within the localization problem, typically, vision sensors are used. It can be seen that the monocular camera is commonly used for this task [20]. Compared to this work, Yoneda et al. [26] utilized LIDAR to localize the vehicle in 3D point clouds. However, the precise LIDAR used in their work is expensive and not feasible for practical applications. In spite of the recent advancements in vehicle localization techniques, it can be seen that the problem still remains a challenge. Recently, John et al. [15] utilize the particle swarm optimization to localize the vehicle in the point cloud using stereo vision. However, the standard particle swarm optimization algorithm is shown to be computationally expensive.

We propose to localize the vehicle in the dense point cloud map using stereo depth information and a novel multiswarm particle swarm optimization algorithm. Furthermore, novel bootstrapping phase is introduced to stabilize the GPS information prior to the vehicle localization. Our solution with the stereo vision reduces the overall cost of the system and can also achieve a good performance for vehicle localization.

3 Localization algorithm

We implement vehicle localization in the point cloud map using stereo-based depth information. Our experimental vehicle is equipped with a GPS, an odometer, and a stereo vision system. The point cloud maps are generated by the mobile mapping system (MMS) [1] in advance and contain

detailed 3D information of the environment. The MMS generates the point clouds where each 3D data point contains the latitude, longitude, and altitude information. The latitude and longitude information is represented in the 2D Universal Transverse Mercator (UTM) system.

Given a pre-defined point cloud map, the autonomous vehicle can be localized within the point cloud using GPS information. However, the GPS information is prone to intermittent missing signal in certain areas. Thus, in our proposed work, we perform the vehicle localization using the stereo-based depth information. To perform the localization, we generate virtual depth maps from the 3D point clouds using a set of transformation matrices. The set of three transformation matrices perform the transformation between three coordinate systems. The first matrix transforms the point cloud maps from the world to the vehicle coordinate T_w^v (world vehicle). The second matrix transforms these transformed data points from the vehicle coordinate to stereo coordinate T_v^s (vehicle stereo). Finally, using the intrinsic parameters of the stereo camera, T_s^i , the virtual depth map is generated. Among the three matrices, the vehicle stereo T_v^s and stereo intrinsic T_s^i matrices are constant for the autonomous vehicle. While the world-vehicle matrix T_w^v changes as the vehicle moves. The *world coordinate system* is defined at the origin of the UTM coordinate. The *vehicle coordinate system* corresponds to the GPS location in the vehicle, and the *stereo coordinate system* corresponds to the left camera in the stereo rig. An overview of the three coordinate systems is provided in Fig. 1a.

We estimate the optimal set of transformation matrices in three phases and perform the vehicle localization. The three phases are represented by the off-line, bootstrapping, and online phases. In the off-line phase, the fixed matrix T_s^i is obtained by calibration, while the fixed vehicle stereo matrix T_v^s is estimated using the PSO. In the bootstrapping phase, we obtain an initial estimate of the time-varying matrix T_w^v using the Kalman filter and GPS-INS information. The initial estimated of world-vehicle matrix is used to initialize a novel multiswarm PSO-based tracker in the online phase. In the PSO-based optimization framework, candidate transformation matrices are used to estimate generate candidate virtual depth maps. These candidate virtual depth maps are evaluated within a depth-based cost function. The cost function evaluates the “goodness of fit” between the candidate virtual depth maps M_v , generated by the PSO, and the online stereo depth map $M_s(t)$. By estimating the optimal set of candidate transformation matrix, we perform the vehicle localization. To reduce the computational complexity of the PSO evaluation, we prune the stereo depth map using the V-disparity method proposed by Long et al. [18]. The depth evaluations are confined to the remaining regions of the stereo depth map. We next review the different algorithm phases. An overview of the algorithm is presented in Fig. 1b.

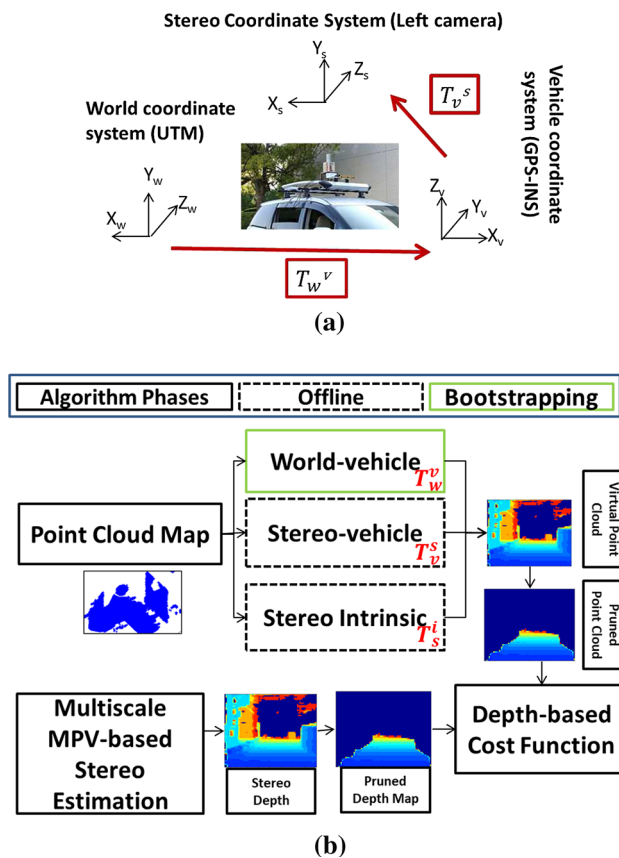


Fig. 1 An illustration of the a different coordinate systems and b the optimization-based localization framework

3.1 Off-line phase

In the off-line phase, we estimate the fixed vehicle stereo, T_v^s , and stereo intrinsic matrix, T_s^i . The matrix T_s^i corresponds to the intrinsic calibration parameter of the vehicle’s stereo rig and is obtained by calibrating the stereo rig using a checkerboard pattern. To estimate the fixed vehicle stereo matrix, we acquire a dataset using our experimental vehicle in a pre-defined area where the time-varying transformation matrix T_w^v information is readily available. More specifically, the dataset is acquired in areas without tall buildings and good satellite coverage. In such areas, the GPS-INS information directly corresponds to the varying transformation matrix information with minimal errors.

Given the calibrated T_s^i matrix and acquired GPS-INS-based world-vehicle T_w^v matrix, we utilize the PSO algorithm to estimate the fixed matrix T_v^s . The PSO algorithm estimates an optimal T_v^s by generating and evaluating a set of candidate virtual depth maps. The set of virtual depth maps are generated by PSO using the corresponding set of candidate transformation matrices. These set of virtual depth maps are then evaluated by the PSO by measuring their similarity with the stereo depth map. The stereo depth map is generated from

the vehicle using the MPV algorithm [18]. To enhance the computational efficiency of PSO evaluation, we prune the depth map using a disparity map-based algorithm. The estimated depth map is used in the PSO cost function to evaluate the virtual depth maps. Using the complete stereo map for evaluation is computationally expensive. Thus, we prune the depth map and limit the evaluation to only certain depth map regions. More specifically, we utilize the curb detection algorithm proposed by Long et al. [18] to remove objects such as pedestrians and vehicles from the depth map.

The PSO generates candidate virtual depth maps using the candidate vehicle stereo transformation matrix T_v^s . More specifically, each particle in the PSO swarm, \mathbf{x}_m , is a vector representation of the candidate matrix T_v^s . The m th particle is given as $\mathbf{x}_m = [e_x, e_y, e_z, \theta, t_x, t_y, t_z]$, where e_x, e_y, e_z, θ represents the axis-angle representation of the rotation matrix and t_x, t_y, t_z represents the translation parameters.

The candidate virtual depth maps generated by the PSO particles are evaluated by the PSO cost function. The PSO cost function measures the similarity between the candidate virtual depth map and the stereo depth map generated by the MPV algorithm. To reduce the computational efficiency, the depth maps are pruned and the PSO evaluation is restricted to the unpruned regions. The cost function for the PSO evaluation is obtained by measuring the Euclidean distance between the candidate pruned virtual depth maps and the pruned stereo depth maps. This is given as:

$$f(\mathbf{x}') = \text{dist}(M_s, M_v(\mathbf{x}', T_s^i, T_w^v)) \quad (1)$$

where \mathbf{x}' represents the PSO particle, which is a vector representation of the candidate transformation matrix T_v^s . Note that T_s^i and T_w^v matrices are available for the vehicle stereo matrix estimation. A detailed overview of the off-line phase is presented in Fig. 2a.

3.2 Bootstrapping phase

Following the estimation of the parameters of the vehicle stereo T_v^s matrix, the vehicle localization is performed by estimating the time-varying world-vehicle transformation matrix T_w^v in the online phase using a PSO-based tracking algorithm. In the bootstrapping phase, we initialize the tracking algorithm using GPS-INS information derived from a bootstrapping sequence with b frames. The GPS-INS directly provides the estimates of the T_w^v matrix for the bootstrapping frames. However, these estimates are not stable. Consequently, we stabilize these estimates over the bootstrapping sequences using the Kalman filter. Following the stabilization, we use the optimal Kalman state parameters $\bar{\mathbf{x}}(b)$ of the b th frame to initialize the PSO tracker. The $\bar{\mathbf{x}}(b)$ is a 7-dim containing the axis-angle representation and translation parameters of T_w^v . We next explain in detail the online

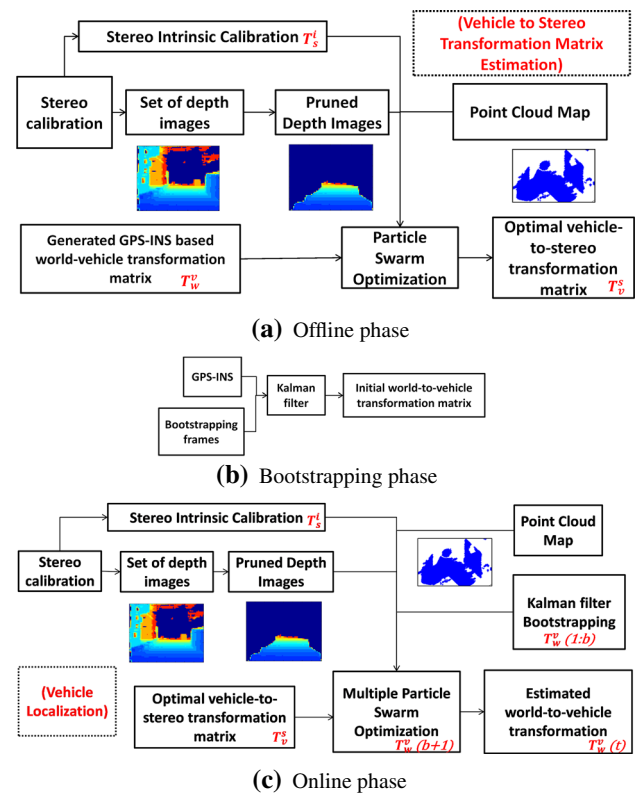


Fig. 2 A detailed overview of the different phases involved in the proposed algorithm

phase. A detailed overview of the bootstrapping phase is presented in Fig. 2b.

3.3 Online phase

We perform the vehicle localization at each time instant t using the novel PSO-based tracker. Given the estimated transformation matrices, i.e., T_v^s and T_s^i , we estimate the remaining time-varying world-vehicle transformation matrix T_w^v without using the GPS-INS information. Typically, variants of the Kalman filter and particle filters are used for the tracking algorithm. However, both these algorithms are prone to divergence. To address this issue, we utilize the PSO algorithm to perform tracking, by estimating the parameters of the T_w^v matrix or $\mathbf{x}_w^v(t)$ at every t frame. Candidate parameters generated by the PSO algorithm, along with the previously estimated fixed transformation matrices, are used to generate the candidate virtual point cloud depth maps. By measuring the similarity between the candidate and stereo depth images, the optimal world-vehicle transformation parameters are estimated and the vehicle is localized. However, the time-varying PSO algorithm is computationally expensive and not suited for online tracking.

The time-varying PSO algorithm [25] functions as a global-to-local optimizer with varying inertia values. At high

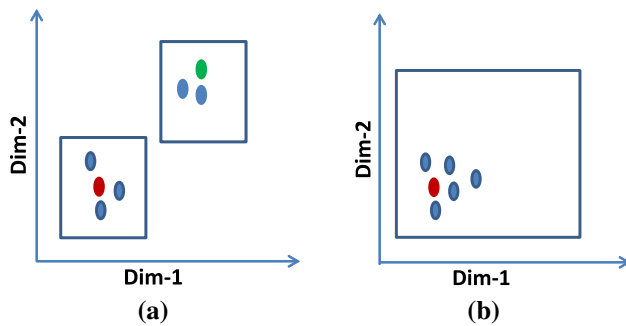


Fig. 3 An illustration of the **a** multiple swarm PSO and **b** single swarm PSO algorithm in a 2-dim search space. The blue bounding boxes represent the search limits, and the blue circles represent the particles. The red circle represents the PSO estimate in the previous frame, while the green circle represents the predicted particle (colour figure online)

inertia values, the PSO functions as a global optimizer, while at lower inertia values the PSO functions as a local optimizer. We propose to localize the vehicle by tracking the Kalman initialized transformation parameters in the online phase. This tracking is proposed as a local constrained search at each frame using the PSO. This is achieved by modifying the global-to-local PSO optimizer to a local optimizer. To account for vehicle motion and increase robustness, we propose to use multiple PSO swarms to perform constrained local search at two locations of the search space. For a given frame t , we initialize the first PSO swarm at the previous frame's $t - 1$ estimate $\mathbf{x}_w^v(t - 1)$. The second swarm is initialized at the predicted estimate for frame t , $\hat{\mathbf{x}}_w^v(t)$. The predicted estimate at frame t is generated using the velocity vector $\mathbf{v}(t - 1)$ by,

$$\mathbf{v}(t - 1) = \mathbf{x}_w^v(t - 2) - \mathbf{x}_w^v(t - 1) \quad (2)$$

The PSO swarm at both the locations are initialized using a pre-defined diagonal covariance Φ_b . Following the initialization, each swarm performs the local search independently following the PSO algorithm. At the end of PSO iteration, the best global particles of the two swarms are compared and the global particle with better cost function value is selected as the estimate for frame t . Examples of the localization results are shown in Fig. 4. Henceforth, we refer to this algorithm as the multiple PSO multiple search limit (MPML). An illustration of the proposed MPML is shown in Fig. 3. Similar to the off-line PSO evaluation, the candidate generated by the multiswarm PSO is also evaluated by the depth-based measure. A detailed overview of the online phase is presented in Fig. 2c.

4 Experimental results

The proposed algorithm is validated on multiple acquired data sets. We performed a detailed comparative and paramet-

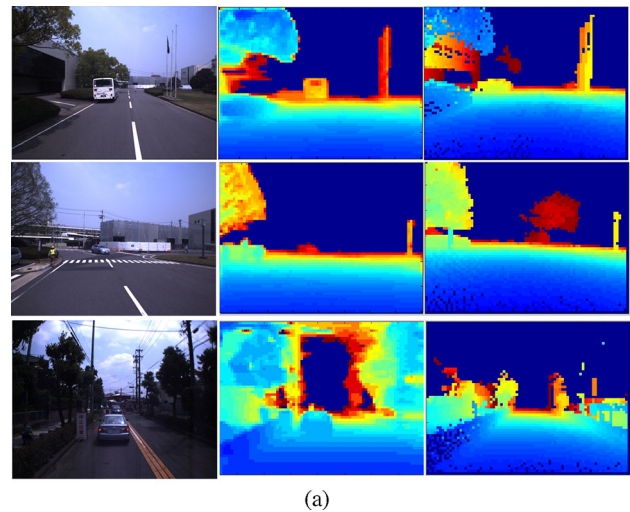


Fig. 4 Localization results from the online phase on the acquired dataset using the MPML algorithm. The left column denotes the left image from the stereo pair; the middle column denotes the estimated disparity. The right column represents the estimated point cloud virtual depth image

ric analysis of the algorithm. Different variations in the PSO tracker are also evaluated. We perform two sets of experiments. In the first set, we evaluate the estimation of the vehicle stereo coordinate transform by the off-line PSO algorithm. In the second set, we evaluate the vehicle localization by the online multiswarm PSO algorithm. Our algorithm was implemented on a Windows machine (3.5 GHz Intel i7 processor) with MATLAB.

Data sets and algorithm parameters The training data set for the off-line phase contains 15 stereo depth maps and corresponding T_w^v parameters derived from the GPS-INS sensor. We used 5 PSO particles, c_1 and c_2 were set at 2, A was set to 0.5, and C was set to 100 PSO iterations.

We acquired three data sets with multiple sequences for the online phase. The first data set contains 4 sequences with 300 frames. The second data set contains 5 sequences with 1200 frames. The third data set contains 3 sequences with 1200 frames. The length of the bootstrapping sequence b was set to 5. The MPML contained 5 particles with the previous estimate swarm containing 3 particles and the predicted estimate swarm containing 2 particles. For both the swarms, C was set to 50 iterations and A was set to 0.1 to facilitate a local search. Finally, we define the search limits for the PSO search, based on the maximum inter-frame velocity.

4.1 Estimation of the vehicle stereo coordinate transform

In our first experiment, we evaluate the off-line phase of our proposed algorithm. More specifically, we evaluate PSO's estimation of the fixed transformation matrix T_v^s . We perform a comparative analysis of the PSO algorithm with the

baseline algorithms, including genetic algorithm (GA) and the simulated annealing algorithm (SA) [10]. To facilitate a fair comparison, the algorithm parameters of the GA and SA were kept similar to the PSO. Firstly, the number of generations and iterations of the GA and SA is equal to the PSO iterations. Secondly, the same search limits were also used to constrain the GA and SA. We report the Euclidean distance measure between the estimated transformation matrices and the ground truth transformation matrix parameters. The errors reported in our results correspond to the distances in the transformation matrix-based “feature space.” As discussed earlier, these transformation matrices are represented in 7-dim vector form containing the axis–angle representation and the translation parameters. The ground truth parameter for this evaluation is obtained from the manually calculated distance and orientation between the GPS (vehicle coordinate) and stereo camera (stereo coordinate) on the experimental vehicle.

As shown in Table 1, we observed that the PSO algorithm reported better computational accuracy than the baseline algorithms. Comparing the computational efficiency, we observed that the PSO is computationally efficient. The variations in computational time can be attributed to the search limits enforcement scheme in the algorithms. While the PSO enforces the search limits by simply reversing the sign or search direction of the velocity components, the GA and SA perform re-sampling to enforce the search limits. Thus, the increased computational time can be attributed to the re-sampling scheme in the baseline algorithm.

We also compared two cost functions, which are based on *pruned depth map* and complete stereo depth map, respectively. Table 2 shows that the pruned depth map improved both the accuracy and the computational efficiency.

Table 1 Mean and variance of the Euclidean distance-based “feature space” error measures for the PSO, GA, and SA over 3 trials on the off-line dataset

Algorithms	PSO	GA	SA
Error	0.62 ± 0.08	1.52 ± 0.6	1.32 ± 0.5
Time taken (min)	35	37.25	69.75

Table 2 Mean and variance of the Euclidean distance-based “feature space” error measures for the PSO over 3 trials with and without pruned depth map

Number of iterations	Pruned depth map	Complete stereo
Error	0.62 ± 0.08	1.06 ± 0.2
Time taken (min)	35	37

4.2 Vehicle localization

We evaluate the performance of the multiswarm PSO algorithm. More specifically, we evaluate PSO’s estimation of the varying transformation matrix T_w^v . We report the Euclidean distance measure between the estimated transformation matrices and the ground truth transformation matrix parameters. The errors reported in our results correspond to the distances in the transformation matrix-based “feature space.” As discussed earlier, these transformation matrices are represented in 7-dim vector form containing the axis–angle representation and the translation parameters. The ground truth parameter for the online phase for the tracking sequences corresponds to the GPS–INS parameters. We perform a detailed comparative and parameter analysis of the vehicle localization. For *comparative analysis*, we evaluated the proposed MPML method with the particle filtering framework, i.e., the particle filter (PF) [7] and the annealed particle filter (APF) [8]. Similar to the off-line phase, the particle filtering evaluations were kept similar to the 250 iterations of the MPML. The PF contains 250 particles, and the APF contains 125 particles and 2 layers.

Two sub-experiments were conducted in the comparative study to estimate the T_w^v parameters. In the first sub-experiment, we estimated the $T_w^v(t)$ for all the $t > b$ frames, where b corresponds to the bootstrapping frames. Note that in this experiment, the GPS–INS-based Kalman filter was only used in the bootstrapping frames. The localization in the remaining frames is based on depth only. As shown in Table 3, the proposed algorithm outperformed the baselines. The divergence error in PF caused the low accuracy in the final results.

In the second sub-experiment, termed as the “missing GPS experiment,” we evaluated the localization errors for the intermittently missing GPS signals, by sequentially performing the tracking and re-initialization. More specifically, for a sequence of length n , we performed the bootstrap in the first b frames, followed by tracking in the next k frames. At the end of the $b + k$ frames, a Kalman filter-based re-initialization was run for b frames using the GPS information. The re-initialization is similar to the bootstrapping phase. The optimal Kalman state estimate at the end of the re-initialization phase was adopted to initialize the tracking for

Table 3 Mean and variance of the Euclidean distance-based “feature space” error measure for the first sub-experiment

Dataset	MPML	PF	APF
1	0.23 ± 0.01	0.89 ± 0.27	1.57 ± 0.7
2	0.08 ± 0.01	1.3 ± 0.53	2.04 ± 1.0
3	0.5 ± 0.06	1.1 ± 0.3	2.3 ± 1.6

Table 4 Mean and variance of the Euclidean distance-based “feature space” error measure for the missing GPS experiment with 5 missing frames

Dataset	MPML	PF	APF
1	0.03 ± 0.01	0.32 ± 0.01	0.48 ± 0.03
2	0.01 ± 0.01	0.2 ± 0.01	0.44 ± 0.03
3	0.01 ± 0.02	0.22 ± 0.1	0.37 ± 0.03

the next k frames. Thus, we sequentially performed the tracking and re-initialization in every k and b frames, respectively. The experiments were carried out for $b = 5$ and $k = 5$ frames. The results are tabulated in Table 4, where we can see that the performance improves for smaller intermittent signal.

Different variants of the PSO swarm For the parameter evaluation of the algorithm, we evaluated different variants of the PSO algorithm for comparative study in the experiments. Apart from our proposed algorithm, where 2 independent PSO swarms were initialized to perform the tracking, we also performed an evaluation with the global-to-local PSO algorithm. The PSO algorithm was initialized with 5 particles, A as 0.1 and $C = 50$. A single global search limit was defined accordingly. During the online tracking, for each frame t , we initialized the swarm and search limits around the previous frame’s $t - 1$ estimate. We refer to this algorithm as the single PSO with single search limit (SPSL). The search limit range for this variant was set up to be bigger than the MPML’s search limit range to account for inter-frame motion. The results obtained with variants are given in Table 5 for the localization experiments and in Table 6 for the missing GPS experiment. It can be seen that the MPML achieves a better performance than the original algorithm. An illustration of the variants is provided in Fig. 3.

Table 5 Mean and variance of the Euclidean distance-based “feature space” error measure for the localization experiment

Dataset	MPML	SPSL
1	0.23 ± 0.01	0.63 ± 0.17
2	0.08 ± 0.01	0.69 ± 0.10
3	0.5 ± 0.06	0.89 ± 0.06

Table 6 Mean and variance of the Euclidean distance-based “feature space” error measure for the missing GPS experiment with 5 missing frames

Dataset	MPML	SPSL
1	0.03 ± 0.01	0.27 ± 0.06
2	0.01 ± 0.01	0.06 ± 0.04
3	0.01 ± 0.02	0.03 ± 0.02

5 Discussion

Based on the experimental results, we found that the stereo vision was a suitable alternative to the expensive LIDAR for depth-based localization. In the off-line phase, the PSO algorithm reported a better performance and computational efficiency than comparative optimization algorithms like the GA and the SA algorithm. This result reflected previous observations on different optimization problems like tracking [13] and registration [14], etc. The better performance can be attributed to the PSO algorithm’s search mechanism. On the other hand, higher computational efficiency is achieved thanks to the search limit mechanism. Unlike the GA and SA which enforce the search limits by re-sampling [14], the PSO enforces the search limits by merely reversing the velocity vector and assigning the boundary values to the particles. While the PSO reported good results for the off-line phase, the computational efficiency was not suitable for the online phase as the time-varying PSO algorithm takes 135 s per frame on a CPU. By utilizing the locally constrained MPSO algorithm, we reported a computational time of 400 ms per frame on CPU. A similar computational time was also reported by the SPSL algorithm, but the parameter estimation accuracy is slightly inferior as observed in Table 5. This can be attributed to SPSL algorithm’s bigger search limits, compared to the MPML (Fig. 4).

6 Conclusion

We propose to localize the vehicle in the dense point cloud using stereo vision. An optimization framework using the particle swarm optimization is formulated to localize the vehicle in autonomous driving. Candidate virtual depth maps are generated from the point cloud using a series of transformation matrices. The optimal set of transformation matrices are searched with a depth-based cost function within the particle swarm optimizer. The proposed algorithm optimizes these transformation matrices in three phases, i.e., the off-line phase, bootstrapping phase, and the online phase. The fixed transformation matrices are estimated in the off-line phase using the particle swarm optimization, while the varying transformation matrix is initialized in the bootstrapping phase and estimated in the online phase using a novel optimizer-based tracking framework. The proposed algorithm is validated with the acquired data sets. In addition, we performed a detailed comparative and parametric analysis. The proposed algorithm reports a better localization accuracy in comparison with the baseline algorithms. In our future work, the algorithm will be implemented on vehicle with GPU and tested extensively with large-scale data sets.

References

1. Aisan Technology Co Ltd: What's MMS. <http://www.whatmms.com/> (2013). Accessed 7 Nov 2018
2. Aulinas, J., Petillot, Y., Salvi, J., Lladó, X.: The SLAM problem: a survey. In: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence, IOS Press, Amsterdam, The Netherlands, pp. 363–371 (2008)
3. Bazeille, S., Battesti, E., Filliat, D.: Qualitative localization using vision and odometry for path following in topo-metric maps. In: Proceedings of the 5th European Conference on Mobile Robots, Örebro, Sweden, pp. 303–308 (2011)
4. Cappelle, C., Pomorski, D., Yang, Y.: Gps/ins data fusion for land vehicle localization. In: The Proceedings of the Multiconference on “Computational Engineering in Systems Applications”, pp. 21–27 (2006)
5. Colombo, O.: Ephemeris errors of GPS satellites. *Bull. Godsiq* **60**(1), 64–84 (1986)
6. Cunha, J., Pedrosa, E., Cruz, C., Neves, A.J.R., Lau, N.: Using a depth camera for indoor robot localization and navigation. In: In Robotics Science and Systems (RSS) RGB-D Workshop, pp. 1–6 (2011)
7. Dailey, M.N., Parnichkun, M.: Simultaneous localization and mapping with stereo vision. In: Ninth International Conference on Control, Automation, Robotics and Vision, Singapore, pp. 1–6 (2006)
8. Deutscher, J., Blake, A., Reid, I.D.: Articulated body motion capture by annealed particle filtering. In: Conference on Computer Vision and Pattern Recognition, Hilton Head, USA, pp. 2126–2133 (2000)
9. El Jaafari, I., El Ansari, M., Koutti, L.: Fast edge-based stereo matching approach for road applications. *Signal Image Video Process.* **11**(2), 267–274 (2017)
10. Franconi, L., Jennison, C.: Comparison of a genetic algorithm and simulated annealing in an application to statistical image reconstruction. *Stat. Comput.* **7**(3), 193–207 (1997)
11. Grewal, M.S., Weill, L.R., Andrews, A.P.: *Global Positioning Systems, Inertial Navigation, and Integration*. Wiley, New York (2007)
12. Jia, B., Feng, W., Zhu, M.: Obstacle detection in single images with deep neural networks. *Signal Image Video Process.* **10**(6), 1033–1040 (2016)
13. John, V., Trucco, E., Ivekovic, S.: Markerless human articulated tracking using hierarchical particle swarm optimisation. *Image Vis. Comput.* **28**(11), 1530–1547 (2010)
14. John, V., Long, Q., Liu, Z., Mita, S.: Automatic calibration and registration of lidar and stereo camera without calibration objects. In: IEEE International Conference on Vehicular Electronics and Safety, ICVES 2015, Yokohama, Japan, November 5–7, 2015, pp. 231–237 (2015)
15. John, V., Long, Q., Xu, Y., Liu, Z., Mita, S.: Registration of GPS and stereo vision for point cloud localization in intelligent vehicles using particle swarm optimization. In: *Advances in Swarm Intelligence—8th International Conference, ICSI, Fukuoka, Japan, Proceedings, Part I*, pp. 209–217 (2017a)
16. John, V., Tsuchizawa, S., Liu, Z., Mita, S.: Fusion of thermal and visible cameras for the application of pedestrian detection. *Signal Image Video Process.* **11**(3), 517–524 (2017b)
17. John, V., Liu, Z., Mita, S., Guo, C., Kidono, K.: Real-time road surface and semantic lane estimation using deep features. *Signal Image Video Process.* **12**(6), 1133–1140 (2018)
18. Long, Q., Xie, Q., Mita, S., Tehrani, H., Ishimaru, K., Guo, C.: Real-time dense disparity estimation based on multi-path viterbi for intelligent vehicle applications. In: *British Machine Vision Conference, BMVC 2014, Nottingham, UK*, pp. 1–8 (2014)
19. Maier, D., Hornung, A., Bennewitz, M.: Real-time navigation in 3d environments based on depth camera data. In: *12th IEEE-RAS International Conference on Humanoid Robots, Osaka, Japan*, pp. 692–697 (2012)
20. Mattern, N., Schubert, R., Wanielik, G.: High-accurate vehicle localization using digital maps and coherency images. In: *IEEE Intelligent Vehicles Symposium, La Jolla, CA, USA*, pp. 462–469 (2010)
21. Niknejad, H.T., Takeuchi, A., Mita, S., McAllester, D.: On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation. *IEEE Trans. Intell. Transp. Syst.* **13**, 748–758 (2012)
22. Nistér, D., Naroditsky, O., Bergen, J.R.: Visual odometry. In: *2004 Conference on Computer Vision and Pattern Recognition, Washington, DC, USA*, pp. 652–659 (2004)
23. Oh, S.M., Tariq, S., Walker, B.N., Dellaert, F.: Map-based priors for localization. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan*, pp. 2179–2184 (2004)
24. Panzieri, S., Pascucci, F., Ulivi, G.: An outdoor navigation system using GPS and inertial platform. *IEEE/ASME Trans. Mechatron.* **7**(2), 134–142 (2002)
25. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: *International Conference on Evolutionary Computation*, pp. 69–73 (1998)
26. Yoneda, K., Tehrani, H., Ogawa, T., Hukuyama, N., Mita, S.: Lidar scan feature for localization with highly precise 3-d map. In: *2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, June 8–11, 2014*, pp. 1345–1350 (2014)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.