

Visual Odometry based on Random Finite Set Statistics in Urban Environment

Feihu Zhang, Guang Chen, Hauke Stähle, Christian Buckl, Alois Knoll

Abstract—This paper presents a novel approach for estimating the vehicle’s trajectory in complex urban environments. In previous work, we presented a visual odometry solution that estimates frame-to-frame motion from a single camera based on Random Finite Set (RFS) Statistics. This paper extends that work by combining the stereo cameras and gyroscope sensor. We are among the first to apply RFS statistics to visual odometry in real traffic scenes. The method is based on two phases: a preprocessing phase to extract features from the image and transform the coordinates from the image space to vehicle coordinates; a tracking phase to estimate the ego-motion vector of the camera. We consider features as a group target and use the Probability Hypothesis Density (PHD) filter to update the overall group state as the motion vector. Compared to other approaches, our method presents a recursive filtering algorithm that provides dynamic estimation of multiple-targets states in the presence of clutter and high association uncertainty.

The experimental results show that this method exhibits good robustness under various scenarios.

I. INTRODUCTION

USING cameras for vehicle navigation is the current trend in the field of intelligent vehicles. Visual odometry (VO) is gaining importance for estimating vehicle’s trajectory. The main idea is on finding features and matching them between successive images. Based on this information, the motion of the vehicle, the ego-motion, can be calculated. However, there are still open issues in achieving highly robust ego-motion estimation in real traffic scenes as discussed in [1]:

- Features that are used to estimate the ego-motion vector may contain some false associated pairs. Robust matching techniques are needed to avoid false matching.
- The algorithms for ego-motion are typically based on features of stationary objects. However, typical road scenes may contain a large amount of features stemming from moving objects. All these artifacts reduce the performance of ego-motion estimation. An approach is required to estimate the features’ position by utilizing the posterior density to reduce the influences of features from non-stationary objects.

Many of these challenges were addressed in the Structure-from-Motion method [2] which focuses on the removal of the false features. It uses RANSAC [1] to ignore large number of outliers. However, the method does not provide robust behavior in all situations.

Feihu Zhang, Guang Chen, Hauke Stähle and Alois Knoll are with the Technische Universität München, Garching bei München, Germany, e-mail: feihu.zhang@tum.de, {guang,staehle,knoll}@in.tum.de.

Christian Buckl is with fortiss GmbH, München, e-mail: buckl@fortiss.org.

In this paper, we extend existing methods by using Probability Hypothesis Density (PHD) filter to increase robustness. PHD filter works on sets of features, called set-valued states, instead of single features. The observations associated with the features are treated as set-valued observations. Modeling set-valued states and set-valued observations as Random Finite Sets allows to solve the problem of dynamically estimating multiple-targets in the presence of clutter and association uncertainty in a Bayesian filtering framework [3].

In our earlier work [4], we presented a VO system based on a single camera to provide localization information in urban environments. The system estimates the ego-motion vector by utilizing the SIFT [5] (Scale Invariant Feature Transform) features which are calculated from consecutive frames. The tracking algorithm does only take into account two consecutive frames.

This paper enhances our previous system by using a well calibrated stereo camera and a gyroscope sensor. We rely on PHD filtering to track features as many frames as possible and then estimate the ego-motion vector. Additionally, we use SURF [6](Speeded Up Robust Features) instead of SIFT to satisfy real-time constraints.

Our method consists of two phases: a preprocessing phase and a tracking phase. The preprocessing phase starts by extracting features using SURF from the stereo frames and matching them as feature pairs from consecutive frames. Then, the coordinates of the features are transformed to 3D vehicle coordinates. Finally, we record features’ positions as measurements for the tracking phase.

The tracking phase is performed in spatial dimension. In this phase the algorithm tracks features in vehicle coordinates by using the PHD filter to estimate the ego-motion vector at each frame.

We use an off-the-shelf platform providing data under real traffic scenarios [7] to evaluate the approach. The platform offers data from GPS, stereo cameras and gyroscope sensors. The experimental results indicate that the proposed method yields precise estimation.

The benefits of our approach are as follows: First, it eliminates the false matching features since the PHD filter does not need to focus on the data association problem. Second, the PHD filter can utilize the posterior density which is calculated in the Bayesian filtering framework to reduce the influences of non-stationary objects.

The remainder of this paper is structured as follows: Sec. II describes the related work in visual odometry. Sec. III introduces details about the preprocessing phase. Sec. IV de-

scribes the PHD filter and its implementation. Sec. V presents experimental results under real traffic scenes. Finally, the paper is concluded in Sec. VI.

II. RELATED WORK

Much work has been done in visual odometry using e.g. a single camera [8] [9] [10], stereo cameras [11] [12], or an omnidirectional camera [13] [14]. One approach to visual odometry uses the Structure-from-Motion (SfM) technique. The idea is to find good quality features in one frame and the corresponding features in the next frame, estimating displacements from these features and translating them to the motion of the camera [2]. Using RANSAC approach [1] enables the method to overcome a large number of outliers as encountered in real traffic scenes.

Optical flow is a different approach which focuses on the change in the brightness of the image, where this change in brightness results from the apparent motion in the image [15] [16]. This method is much simpler and computationally cheaper than the extraction and tracking of features. However, the precision is not very good. Corke et al.[17] compared these two approaches and got the conclusion that SfM method allows higher precision at the cost of higher computational needs.

In this paper, we apply the SfM technique to estimate the displacements of the vehicle within the RFS framework.

III. PREPROCESSING PHASE

This section describes the steps performed in the preprocessing phase.

A. Feature Extraction

In most of the previous work on visual odometry using SfM technique, features are used for establishing correspondences between consecutive frames in a video sequence. Researchers often use approaches based on Harris corner detector [18], Kanade-Lucas-Tomasi detector (KLT) [19], SIFT [5] or SURF [6].

Since the SURF feature descriptor is highly distinctive and allows robust matching under real time requirements, our system considers matched SURF feature pairs in vehicle coordinates as measurements for the PHD filter (the resulting stereo matches between the first two stereo images are then similarly matched with the stereo matches in the next stereo image).

B. Transformation between Image Coordinates and Vehicle Coordinates

In this step, we determine the mapping between the image coordinates $[u, v]^T$ of a tracked pixel and the vehicle coordinates $[x, y, z]^T$ of the corresponding point. First, we introduce the used coordinate systems (see Fig. 1a). The vehicle coordinates $[x, y, z]^T$ are defined in a three-dimensional cartesian coordinate system with origin in the middle of the rear axle; the camera coordinate $[x_c, y_c, z_c]^T$ are described in a three-dimensional cartesian coordinate system with origin in the optical center of the camera, and the image coordinates

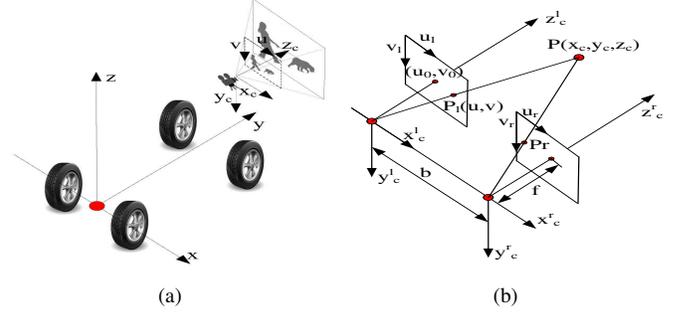


Figure 1. Different coordinates systems

$[u, v]^T$ are defined in a two-dimensional cartesian coordinate system with origin in the upper left corner of the image. For an easy implementation of the transformation, we will use homogeneous coordinates in the following.

1) *Transformation from Vehicle Coordinates to Camera Coordinates:* Considering the object in the image as a rigid body, the transformation from the vehicle coordinates to camera coordinates is represented by six independent extrinsic parameters. These are the three translation parameters within the translation vector $T = [x_t, y_t, z_t]^T$, and the three rotation parameters within the rotation matrix $R = R_x R_y R_z$. R is the rotation matrix corresponding to the product of rotations around the z-axis (rotation matrix R_z), y-axis (rotation matrix R_y) and x-axis (rotation matrix R_x). In summary, the transformation between vehicle and camera coordinates is

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (1)$$

2) *Transformation from Camera Coordinates to Image Coordinates:* For stereo camera system, we have two camera coordinates: left camera coordinates $[x_c^l, y_c^l, z_c^l]^T$ and right camera coordinates $[x_c^r, y_c^r, z_c^r]^T$ (see Fig. 1b). For an easy implementation of the transformation we will use the camera coordinate $[x_c, y_c, z_c]^T$ instead of the left camera coordinates $[x_c^l, y_c^l, z_c^l]^T$. We calculate the 3D coordinates of the pixel from both stereo images in the camera coordinates (left camera coordinates). Three-dimensional objects of a scene are projected onto the two-dimensional surface of the camera sensor. From a standardized projection of a point $\mathbf{P} = [x_c, y_c, z_c]^T$ in vehicle coordinates system based on pinhole model, we can get its image coordinates $[u, v]^T$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{f x_c}{z_c} + u_0 \\ \frac{f y_c}{z_c} + v_0 \end{bmatrix} \quad (2)$$

where (u_0, v_0) is the point of camera optical axis and image plane intersection. This formula can be expressed in homoge-

neous coordinates as

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \quad (3)$$

From equation (1) and equation (3), we can express the relationship between the vehicle coordinates $[x, y, z]^T$ of a point in the space to its image coordinates $[u, v]^T$ as follows:

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (4)$$

Rearranging the above equation, we can transform the pixels from image coordinates to vehicle coordinates.

IV. TRACKING PHASE

A. Overview on RFS Statistics

The Random Finite Set (RFS) is a hidden markov chain model with set-valued states and set-valued observations while the PHD filter is a predict and correct framework for recursive Bayesian filtering in such a RFS formulation. A comparison of the RFS approach and traditional multiple-target tracking methods has been given in [20]. In the PHD filter, the collection of individual targets is treated as set-valued states, and the collection of individual observations is treated as set-valued observations. Fig. 2 is a basic introduction of the PHD filter which shows that the observations and their estimated states are treated as a single valued measurement and its corresponding estimation at each frame [21]. The PHD filter operates on the single-target state space and avoids the combinatorial problem that arises from data association.

The Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter is a closed form implementation of the PHD filter, which is based on the Bayesian estimation framework utilizing random finite sets as the mathematical backbone [22].

Clark [23] introduced a particle PHD filter implementation for tracking group targets. The results of the implementation show a high robustness when tracking group targets in cluttered environments.

B. Kalyan [24] and John. M [25] implemented the PHD filter in the field of simultaneous localization and mapping (SLAM) problem. Results show that the use of a PHD filter is an effective solution to the SLAM problem.

In this paper, we use the PHD filter to estimate the motion of the target set instead of tracking individual targets. Assuming the motion of the whole targets to be equal, we calculate the average of each target's state to acquire the motion of the whole set. From the physical model, we consider the whole set to have the same motion vector with the vehicle. According to this model we estimate the vehicle's ego-motion vector at each frame. Compared to earlier approaches, our work is among the first to apply PHD filtering to visual odometry in real traffic scenes.

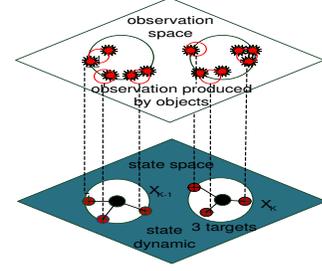


Figure 2. Set-valued states and Set-valued observations

B. Mathematic Background on the PHD Filter

The PHD filter is an approximation to alleviate the computational intractability of the optimal multi-target Bayes filter, proposed by Mahler [26]. Considering the RFS of survived targets $S_{k|k-1}$ between iterations $k-1$ and k , the RFS of spawned targets $B_{k|k-1}$ and the RFS of spontaneous birth targets σ_k , the global RFS characterizing the multiple-target set can be written as:

$$\mathbf{X}_k = [\bigcup_{\zeta \in \mathbf{X}_{k-1}} S_{k|k-1}(\zeta)] \cup [\bigcup_{\zeta \in \mathbf{X}_{k-1}} B_{k|k-1}(\zeta)] \cup \sigma_k \quad (5)$$

The set observation \mathbf{Z}_k can be seen as a global RFS composed by the RFS of measurements originally from the targets $\theta_k(\mathbf{x})$ and by the RFS of false alarms κ_k , modeled by a Poisson distribution.

$$\mathbf{Z}_k = [\bigcup_{\mathbf{x} \in \mathbf{X}_k} \theta_k(\mathbf{x})] \cup \kappa_k \quad (6)$$

The PHD filter evolves in two steps: prediction and update. The multiple-target posterior density of the target RFS is also called the intensity function D . The transition function is denoted as $f_{k|k-1}(\mathbf{x}|\zeta)$ given the previous state ζ .

The prediction equation of the PHD filter is:

$$D_{k|k-1}(\mathbf{x}) = [\int [P_S(\zeta) f_{k|k-1}(\mathbf{x}|\zeta) + \beta(\mathbf{x}|\zeta)] D_{k-1}(\zeta) d\zeta] + \gamma_k \quad (7)$$

Knowing the measurement random set \mathbf{Z}_k , it is possible to update the intensity function as follows:

$$D_k(\mathbf{x}) = (1 - P_D) D_{k|k-1}(\mathbf{x}) + \sum_{z \in \mathbf{Z}_k} \frac{P_D g_k(\mathbf{z}_i|\mathbf{x}) D_{k|k-1}(\mathbf{x})}{\kappa(\mathbf{z}_i) + \int P_D g_k(\mathbf{z}_i|\zeta) D_{k|k-1}(\zeta) d\zeta} \quad (8)$$

where P_D is the probability of detection. $g_k(\mathbf{z}_i|\mathbf{x})$ is the likelihood of state \mathbf{x} given an observation \mathbf{z}_i .

The prediction PHD equation (7) includes components whose intensities are affected by targets that enter the scene (γ_k), targets that spawn new targets ($\beta(\mathbf{x}|\zeta)$), and targets that survive from the previous time step P_S . D_{k-1} is the posterior PHD from the previous time step.

The update PHD equation (8) corrects the predicted PHD by including evidence from the current set of observations. Knowledge about scene clutter $\kappa(\mathbf{z})$ is also embedded into the update step. $\kappa(\mathbf{z})$ is the intensity of clutter (expected number of observations arising from the clutter at \mathbf{z}).

$$N(k) = \int_{\Psi} D_{k|k}(\mathbf{x}) d\mathbf{x} \quad (9)$$

Equation (9) illustrates that the integral of the PHD over a certain domain Ψ yields the estimated number of targets $N(k)$ in that domain at time k . The PHD is not a probability density and does not necessarily sum up to 1 [26].

It is to be noted that the PHD recursion involving equations (7) and (8) have multiple integrals that have no closed form solutions in general. One of the common approaches to mitigate this problem is to use GM-PHD approximations. More details and a generic description of the GM-PHD filter are given by [22]

C. Implementation Details

Our algorithm is implemented in vehicle coordinates. We assume that the vehicle is driving on a flat plane. Suppose at frame k , the coordinates of one feature point is (x_k, y_k, z_k) . According to Euler's rotation theorem the relationship between the feature points in consecutive frames is as follows:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} \cos\Delta\beta_k & -\sin\Delta\beta_k & 0 \\ \sin\Delta\beta_k & \cos\Delta\beta_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k - \Delta x_k \\ y_k - \Delta y_k \\ z_k \end{bmatrix} \quad (10)$$

where $(\Delta x_k, \Delta y_k, \Delta\beta_k)$ is the vehicle's ego-motion vector. In this paper, $(\Delta x_k, \Delta y_k)$ is the vehicle's movement in (x, y) direction and $\Delta\beta_k$ is the vehicle's angular change related to the z-axis.

- From the physical model we consider the target has the same motion process. In the same manner like equation (10), we assume that at frame k the target moves independently and the motion process parameters are:

$$\mathbf{x}_k = [x_k, y_k, z_k, \beta_k, \dot{x}_k, \dot{y}_k]^T \quad (11)$$

$$\mathbf{F} = \begin{bmatrix} \cos\beta_k & -\sin\beta_k & 0 & 0 & -\cos\beta_k & \sin\beta_k \\ \sin\beta_k & \cos\beta_k & 0 & 0 & -\sin\beta_k & -\cos\beta_k \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

The process noise is defined by:

$$Q_k = \text{diag}([\sigma_{x1}^2, \sigma_{y1}^2, \sigma_{z1}^2, \sigma_{\beta1}^2, \sigma_{x1}^2, \sigma_{y1}^2]) \quad (13)$$

- The measurement vector is as follows:

$$\mathbf{z} = [x, y, z, \beta]^T \quad (14)$$

where (x, y, z, β) is acquired according to the coordinates transformation process and the gyroscope sensor. To map the state vectors to the observation space, the observation matrix is:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (15)$$

The observation noise is described as:

$$R_k = \text{diag}([\sigma_x^2, \sigma_y^2, \sigma_z^2, \sigma_\beta^2]) \quad (16)$$

- Aggregated targets falling below a given threshold are pruned and the remaining targets are reweighted accordingly (weight is consider as the possibility of the target compared to clutter). If the distance of the targets defined by the covariance matrix falls within a merging threshold τ , then the targets are merged.

From the physical model we consider the whole group set to have the same motion vector. According to this motion model, we calculate the average state of the targets as the ego-motion vector at frame k .

$$\mu_{\mathbf{k}} = [\Delta x_k, \Delta y_k, \Delta\beta_k]^T \quad (17)$$

where $(\Delta x_k, \Delta y_k, \Delta\beta_k)$ is the mean of the state set which is calculated by:

$$\begin{aligned} \Delta x_k &= \frac{1}{N(k)} \sum_{i=1}^{N(k)} \dot{x}_k^i \\ \Delta y_k &= \frac{1}{N(k)} \sum_{i=1}^{N(k)} \dot{y}_k^i \\ \Delta\beta_k &= \frac{1}{N(k)} \sum_{i=1}^{N(k)} \beta_k^i \end{aligned} \quad (18)$$

According to the pruned and merged technology, the PHD filter can keep on tracking the targets as many frames as possible without concerning the data association problem.

First, the PHD filter initialized the new birth targets at current frame by the SURF features (from the stereo image pairs) in vehicle coordinates. Second, it utilizes the measurements at next frame to update the previous estimated results (including the previous birth targets). Finally, it repeats the above steps frame by frame to estimate the motion vector.

Since the PHD filter estimates the whole set's motion vector within the RFS framework, $\mu_{\mathbf{k}}$ is not only used to calculate the trajectory of the vehicle at frame k , but also used to initialize the parameters of the birth models at frame $k + 1$.

The birth models consist of the initial observations (initialized by the features' coordinates and the previous estimated motion vector e.g. equation (17) from consecutive frames). There are no spawn targets in our model.

D. Relationship between PHD filter and Kalman filter

The differences between our approach and visual odometry using Kalman filter (which focuses on tracking the features) are as follows:

First, the Kalman filter is only used to track individual targets. However, it needs to solve the data association problem from consecutive stereo image pairs. The false matching features may influence the estimation while the PHD filter can distinguish the false features as clutters according to the random set statistics. Second, the Kalman filter is neither able to aggregate measurements nor to separate one measurement

to multiple targets. It needs a one-to-one relation between real measurements and expected measurements, solved by data association. A PHD overcomes this, as it is an n-to-m mapping. We calculate the average of each target's state to acquire the whole set's motion vector within the RFS framework.

Since equation (12) is nonlinear, we use the GM-EK-PHD filter to estimate the state which is very similar to the GM-PHD. More details about EK-PHD can be found in [22].

V. EXPERIMENTAL RESULTS

The visual odometry algorithm described in this paper has been implemented on a Core 2 Duo 3.0 Ghz computer. The algorithm is executed in Matlab and the average processing time is 70ms per frame in tracking phase. The vehicle platform is used from Karlsruhe dataset [7] which included GPS, gyro and stereo images at 10 frames/s with a resolution of 1344×391 pixels. All sequences correspond to real traffic conditions in urban environments with pedestrians and other cars. In the experiments, the vehicle was driven with an average velocity of 40km/h.

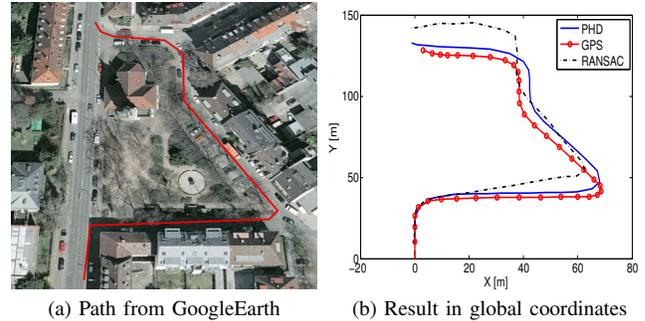
As can be observed, we also use the RANSAC algorithm to estimate the ego-motion vector [27]. A dead reckoning method is used to calculate the trajectory according to the ego-motion vector provided by both the RANSAC algorithm and the PHD filter.

Fig. 3 shows the result of our approach. The left part of the figure shows an aerial view of the area where the experiment was conducted. The right part of the figure illustrates the 2D trajectory estimated by the visual odometry algorithm presented in this paper compared with RANSAC algorithm. From Fig. 3b and Fig. 3c, we can see that our approach provides better accuracy under real traffic scenarios compared to RANSAC.

Fig. 4 compares the number of SURF features and the PHD effective tracking features during the whole process at each frame. From Fig. 4, we can see that the number of SURF features is higher than the number of targets in the set-valued states estimated by the PHD filter. This phenomenon illustrates that during the whole process there must contain certain number of falsely matched feature pairs, which might influence the performance of the visual odometry result. The PHD filter can intelligently choose the effective features to calculate the ego-motion vector. The experiment doesn't compare the numbers of the effective SURF features in RANSAC since the reason of the RANSAC approach keeps a certain percentage of outliers during the whole process.

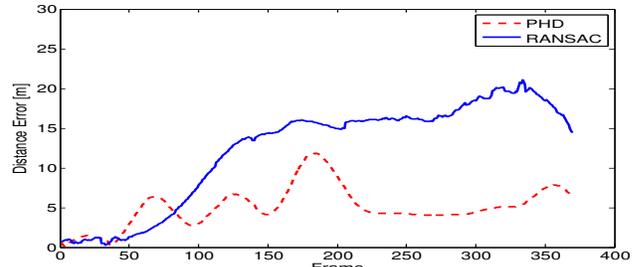
Fig. 5 compares the measurements of the vehicle's angular change from the gyroscope sensor and its estimated results from the PHD filter. It illustrates that the estimated results are more smoothing than the observations while the vehicle is turning. Fig. 3b also shows that when the vehicle is turning the path calculated by visual odometry is more smoothing than GPS.

Our approach was tested in comparison to RANSAC on all tracks provided by Karlsruhe dataset [7]. Our approach lead to a decrease between 25% and 80% on the average RMSE



(a) Path from GoogleEarth

(b) Result in global coordinates



(c) Absolute position error

Figure 3. Visual odometry result

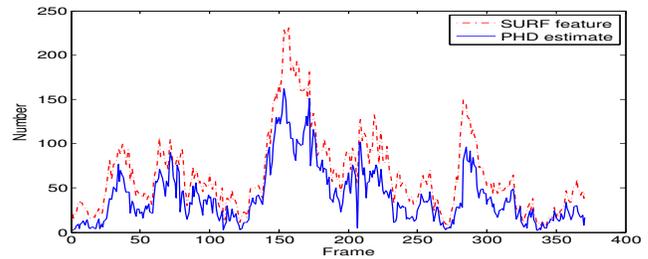


Figure 4. Average number of targets

(root mean square error) compared to RANSAC. The average improvement was 54%.

There are two benefits compared with RANSAC in visual odometry:

- 1) RANSAC has been established as the standard method for motion estimation in the presence of outliers (false features). RANSAC achieves its goal by iteratively selecting a random subset of the original data. However,

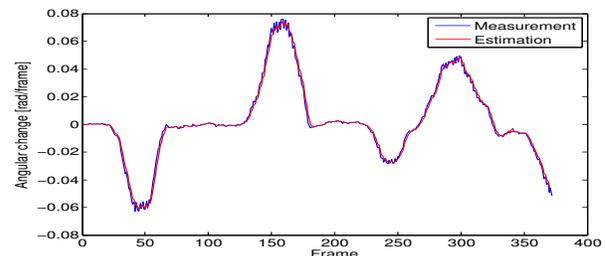


Figure 5. The vehicle's angular change

the PHD filter treats the false features as clutters within the RFS framework by using the dynamics model (physical laws of motion).

- 2) It is difficult for RANSAC to remove outliers when the VO system has a large proportion of features that stem from moving objects. These features can influence the precision of the results. However, within the RFS framework, the PHD filter propagates the posterior intensity, a first-order statistical moment of the posterior multiple-target state at each frame. According to the dynamic system model (equation (12)), it can utilize the estimated state to reduce the influences from those features.

The results of our experiments indicate that the algorithm performs robustly in the presence of pedestrians, vehicles and shadows on the road.

VI. CONCLUSION

Visual odometry in urban scenes is challenging due to a large amount of outliers. The clutter from falsely matched features and moving objects cause the results to deviate from the real status. In this paper, an approach of PHD filtering under RFS framework is presented. In comparison to the other works, this contribution is among the first to apply a PHD filter to visual odometry in a real traffic scenes. With this, visual features are considered as a group target – we then utilize the average of each target’s state to approximate the ego-motion vector since all targets should have the same motion vector in the whole group set within the RFS framework. Compared to other approaches, our approach presents a recursive filtering algorithm that provides dynamic estimation of multiple-targets states in the presence of clutter and high association uncertainty. The evaluation results show that the algorithm achieves high accuracy and robustness under different scenes.

Future improvement of the visual odometry system may use a better movement model which not only estimates the ego-motion vector but also estimates the state of the moving targets in a large scale urban environment.

REFERENCES

- [1] R. Garcia, M. Sotelo, I. Parra, D. Fernandez, and M. Gavilan, “2d visual odometry method for global positioning measurement,” in *IEEE International Symposium on Intelligent Signal Processing, WISP 2007*, Oct. 2007, pp. 1–6.
- [2] D. Scaramuzza, F. Fraundorfer, M. Pollefeys, and R. Siegwart, “Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints,” in *2009 IEEE 12th International Conference on Computer Vision*, Oct. 2009, pp. 1413–1419.
- [3] B.-N. Vo, S. Singh, and A. Doucet, “Sequential monte carlo implementation of the phd filter for multi-target tracking,” in *Proceedings of the Sixth International Conference on Information Fusion*, 2003, pp. 792–799, Vol. 2.
- [4] F. Zhang, H. Staehle, A. Gaschler, C. Buckl, and A. Knoll, “Single camera visual odometry based on random finite set statistics,” Fakultät für Informatik, Technische Universität München, Tech. Rep., 2012.
- [5] D. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, pp. 1150–1157, Vol. 2.
- [6] H. Bay., T. Tuytelaars., and L. V. Gool, “Surf: Speeded up robust features,” in *9th European Conference on Computer Vision*, May 7-13 2006, pp. 404–417.
- [7] B. Kitt, A. Geiger, and H. Lategahn, “Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme,” in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, June 2010, pp. 486–492.
- [8] A. Davison, “Real-time simultaneous localization and mapping with a single camera,” in *International Conference on Computer Vision*, 2003, pp. 1403–1410.
- [9] D. Nister, O. Naroditsky, and J. Bergen, “Visual odometry,” in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004*, July 2004, pp. I-652–I-659, Vol. 1.
- [10] D. Burschka and G. Hager, “V-gps(slam): vision-based inertial system for mobile robots,” in *2004 IEEE International Conference on Robotics and Automation*, May 2004, pp. 409–415, Vol. 1.
- [11] A. Milella and R. Siegwart, “Stereo-based ego-motion estimation using pixel tracking and iterative closest point,” in *IEEE International Conference on Computer Vision Systems, 2006 ICVS ’06*, Jan. 2006, p. 21.
- [12] K. Konolige, M. Agrawal, and J. Sola, “Large-scale visual odometry for rough terrain,” in *In Proceedings of International Symposium on Research in Robotics*, Japan, Nov. 26-29 2007.
- [13] D. Scaramuzza and R. Siegwart, “Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles,” *IEEE Transactions on Robotics*, no. 5, pp. 1015–1026, Vol. 24, Oct. 2008.
- [14] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, “Real-time monocular visual odometry for on-road vehicles with 1-point ransac,” in *IEEE International Conference on Robotics and Automation, ICRA ’09*, May 2009, pp. 4293–4299.
- [15] C. McCarthy and N. Barnes, “Performance of optical flow techniques for indoor navigation with a mobile robot,” in *IEEE International Conference on Robotics and Automation, ICRA ’04*, May 2004, pp. 5093–5098, Vol. 5.
- [16] J. Campbell, R. Sukthankar, and I. Nourbakhsh, “Techniques for evaluating optical flow for visual odometry in extreme terrain,” in *2004 IEEE/RSSJ International Conference on Intelligent Robots and Systems, IROS 2004*, Sept. 2004, pp. 3704–3711, Vol. 4.
- [17] P. Corke, D. Strelow, and S. Singh, “Omnidirectional visual odometry for a planetary rover,” in *IEEE/RSSJ International Conference on Intelligent Robots and Systems, IROS 2004*, Oct. 2004, pp. 4007–4012, Vol. 4.
- [18] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Alvey Vision Conference*, 1988, pp. 147–152.
- [19] C. Tomasi and T. Kanade, “Detection and tracking of point features,” Carnegie Mellon, Tech. Rep., April 1991.
- [20] I. Goodman, R. Mahler, and H. Nguyen, *Mathematics of Data Fusion*. Norwell, MA: Kluwer Academic, 1997.
- [21] B.-N. Vo, “Random finite sets in stochastic filtering,” EEE Department University of Melbourne Australia, Tech. Rep., 2009. [Online]. Available: <http://www.ee.unimelb.edu.au/staff/bv/>
- [22] B.-N. Vo and W.-K. Ma, “The gaussian mixture probability hypothesis density filter,” *IEEE Transactions on Signal Processing*, no. 11, pp. 4091–4104, Vol. 54, Nov. 2006.
- [23] D. Clark and S. Godsill, “Group target tracking with the gaussian mixture probability hypothesis density filter,” in *3rd International Conference on Intelligent Sensors, Sensor Networks and Information, ISSNIP 2007*, Dec. 2007, pp. 149–154.
- [24] B. Kalyan, W. S. Wijesoma, and K. W. Lee, “Fisst-slam: Finite set statistical approach to simultaneous localization and mapping,” *International Journal of Robotics Research*, no. 2, Sept. 2010.
- [25] J. Mullane, B.-N. Vo, M. Adams, and B.-T. Vo, “A random-finite-set approach to bayesian slam,” *IEEE Transactions on Robotics*, no. 2, pp. 268–282, Vol 27, April 2011.
- [26] R. Mahler, “Multitarget bayes filtering via first-order multitarget moments,” *IEEE Transactions on Aerospace and Electronic Systems*, no. 4, pp. 1152–1178, Vol. 39, Oct. 2003.
- [27] M. Zuliani, “Ransac toolbox for matlab,” [web page] <http://www.mathworks.com/matlabcentral/fileexchange/18555>, Nov. 2008.