RL-EKF Feature-Based Visual SLAM with Habitat Simulator

Maryam Valipour

Report Date: 6 April 2021

I. Introduction

THE problem of simultaneous localization and mapping (SLAM) has been studied for a long time, and it is still one of the essential issues in the robotics field. The problem of SLAM is defined as the simultaneous tasks of estimating the robot's pose and building a map of the environment. Solving both of these tasks at the same time is very challenging because they are dependent on each other. That is to say, in order to estimate the robot's pose accurately, a good map of the environment is required, and in return, a proper pose estimate is needed to build the environment model accurately. Therefore, SLAM is the basis for most autonomous navigation systems; it is considered a fundamental problem for autonomous robots and is central to a range of indoor, outdoor, airborne, and underwater applications for both manned and autonomous vehicles.

The SLAM problem has been tackled through many approaches; one method is the application of extended Kalman filters (EKF). This approach consists of two main parts, mainly prediction and correction using environment observation from sensors. Different types of sensors can be utilized in the EKF-SLAM problem, including sonar, laser rangefinders, lidars, and cameras. These are used as on-vehicle sensors and are generated from a platform that moves through the environment. The relative measurement obtained from these sensors can be exploited to estimate both the robot's and the landmarks' positions in the environment. The EKF-SLAM problem is usually based on laser or sonar sensors and solved using artificial landmarks as distinct features, such as barcode reflectors, ultrasonic beacons, visual patterns, etc. However, this approach cannot be used in beacon-free environments. So, vision-based methods which use stable natural landmarks as features are in high demand. This type of SLAM is classified as Visual SLAM (vSLAM). As vSLAM approaches need lots of processing, extracting good visual features is more troublesome than traditional methods. In order to be able to localize the robot's pose and landmarks accurately in the environment, high-level features should be extracted from the images. There are a few methods to achieve this goal, such as Harris and Shi-Tomasi corner detection, scale-invariant feature transform (SIFT), Binary robust independent elementary features (BRIEF), and oriented fast rotated BRIEF (ORB).

II. Literature Review

Recently, different types of cameras have been widely used as a sensor for observing the environment in the SLAM problem, including monocular, stereo, and RGB-D cameras.

Davidson et al. developed the first monocular system for feature-based visual SLAM using an extended Kalman filter (EKF) in 2003, which could operate in real-time in smaller environments such as a desk and was name MonoSLAM. The camera motion with 6 degrees of freedom and the 3D position of the feature points are estimated simultaneously using EKF in this model. Also, map initialization is done by observing a known object in a global coordinate system.

ORB-SLAM2 was developed in 2015, and it does not require GPUs for operation and can provide real-time performance with desktop CPUs. This algorithm is divided into three main parts, including tracking, local mapping, and loop closing. In the first part of the algorithm, tracking, feature matching is done and are compared with the local map to localize the camera in real-time. Also, motion-only bundle adjustment is performed in this section to minimize the reprojection error of each feature. It's also worth mentioning that by only considering the features instead of the whole image, the ORB-SLAM algorithm loses some valuable information. For the next section, local maps are made and optimized with the use of algorithms such as iterative closest point (ICP). Also, local bundle adjustment is executed to optimize the keypoints location for computing the camera's most likely location. For the last part, it corrected the accumulated map drift and executes loop closure. And finally, a full bundle adjustment is performed after loop closure to avoid the map drift and correcting the robot's location. ORB-SLAM2 can be used with different types of cameras, such as monocular, stereo, and RGB-D.

III. Feature Detection and Matching

In feature-based vSLAM problems, landmarks are extracted by examining each pixel to find locally distinct points in the images, which is done by leveraging advanced computer vision techniques in image processing. These specific points in the images are called interest points (key-points), which stick out compared to their local neighborhood and are even locally distinct from different viewpoints. These interest points are illustrated by feature vectors/descriptors, which basically is a compact representation of the local neighborhood around the key-point. The landmarks extracted from the image should possess a few qualities in order to be a reliable choice for the vSLAM problem, such as being stationary, easily re-observable, and distinguishing from other points in the image in different scales, rotations, and lighting of the image.

The fundamental features that are distinguished in keypoint detection techniques include corners, edges, blobs, and ridges, such as mountain peaks or doorways. Corners and blobs are sharp and smooth image features, respectively.

Moreover, the ability to perform data association is an essential part of any feature-based SLAM problem. In the case of VSLAM, matching features and corresponding each measurement to landmarks when re-observed in consecutive image frames as well as in loop closing is much more complicated and is a common problem faced in computer vision. This problem can be tackled through the use of matching methods, such as the k-Nearest Neighbors algorithm and random sample consensus (RANSAC), to detect feature matches between the interest point sets in two or more images. The most important methods to vision feature detection, such as Harris detection, Shi-Tomasi, Scale Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Binary Robust Independent Elementary Features (BRIEF), and Oriented FAST and Rotated BRIEF (ORB), will be investigated in the following paragraphs.

A. Corner Detection using Harris and Shi-Tomasi criteria

The Harris technique was first established in 1988 by Chris Harris and Mike Stephens, which represents an effective way to detect corners in the images. Corners are essential features of images that are comprised of two edges connecting in roughly orthogonal directions. As a result, a significant gradient change is observed in all directions. Moreover, they do not vary with translation, rotation, and illumination. On the other hand, edges are just a sudden change in image brightness, and no gradient change is observed along the edge direction, which makes them not very distinctive and suitable as interest points.

Intensity changes in both X and Y directions should be spotted in order to find corners in an image. This can be done by computing the sum of squared difference (SSD) in image intensity values in a local patch around each point, which is shown in the equation below.

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u,y+v) - I(x,y)]^2 \tag{1}$$

This can be further simplified by applying Taylor expansion:

$$E(u,v) = [u,v]\left(\sum \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}\right)\begin{bmatrix} u \\ v \end{bmatrix} \tag{2}$$

The summed matrix is called the structure matrix, which encodes the variations in image intensities and summarizes the dominant direction of the gradients in a local area around each point. Points are considered as corners if their structure matrix has two large eigenvalues. There are certain criteria for identifying a point as a corner, such as Harris and Shi-Tomasi.

B. Scale Invariant Feature Transform (SIFT)

Corner detector approaches perform well when scale, orientation, viewpoint, and illumination do not change across images. However, in the real world, these conditions do not stand. As a result, there is a need for methods invariant to scale, rotation, illumination, and viewpoint. One such method is called scale-invariant feature transform (SIFT), which was first established by D.Lowe in 2004. At first, a scale-space needs to be generated to guarantee scale invariance. It is produced by progressively blurring out images from the original images and their octaves' using the Gaussian blur operator. At the next step, key-points are found using the generated images' difference of Gaussians (DoG). These SIFT features are located at maxima and minima of the DoG function applied in scale space. Some features found by the SIFT method are considered bad key-points and are not reliable such as edges and low contrast regions, which need to be eliminated. After that, further calculations are done relative to an orientation computed for each key-point, which effectively cancels out the effect of orientation, making this approach rotation invariant. Finally, a feature descriptor is defined by calculating a histogram of gradient orientation information around the local neighborhood of the key-point.

C. Speeded-Up Robust Features (SURF)

This method was first proposed in 2009 by Herbert Bay. SURF is a scale and rotation invariant approach and does both feature extraction and descriptor. It is built on the SIFT method; however, it performs much faster and with higher accuracy. It outperforms the old methods in terms of repeatability, robustness, and distinctiveness of its descriptors. SURF is based on the Fast Hessian matrix, sums of 2D Haar wavelet response, and integral images for image convolution. The determinant of the hessian matrix is computed to select both scale and location in order to achieve higher accuracy and lower computational costs.

D. Binary Robust Independent Elementary Features (BRIEF)

BRIEF was the first binary method, which was initially presented in 2010 by. In BRIEF, binary feature vectors describe key-points by only ones and zeroes in a 128-512 bits string. In this method, at first, a patch is selected around the local neighborhood of each key-point. After that, a set of pixel pairs in that patch is chosen according to specific sampling strategies such as uniform random sampling, Gaussian sampling, etc. For each pair, the image intensities are compared and are concatenated into a bit string. Also, it's worth mentioning that matching is done by computing the hemming distance, which is much faster compared to the L2 norm.

This method is much faster than the previous methods such as SIFT and SURF in both building feature vectors and matching key-points. Furthermore, it is much more efficient in terms of computational costs and can be easily performed in real-time.

E. Oriented FAST and Rotated BRIEF (ORB)

ORB was mainly developed as an alternative to the SIFT and SURF approach because they were patented and could not be used for commercial purposes. However, it performs with much better accuracy comparing to SIFT and SURF in terms of both feature detection and feature matching. Moreover, ORB is much faster than Both SIFT and SURF since it leverages both the Fast approach and binary descriptors such as the BRIEF method. The advantage of ORB over BRIEF is that it is also robust to changes in rotation; in other words, it is rotation invariant.

At first, key-points are detected using the enhanced Fast method that leverages orientation components and multi-scale features. In the fast method, keypoints are detected by considering the brightness of the local neighborhood around each pixel. However, in the ORB approach, a multi-scale aspect is added to the Fast algorithm by using a multi-scale pyramid of the image, which consists of sequences of the image at different resolutions, and then the key-points are detected. After that, an orientation component is added to the patch around the keypoints, which is computed based on the intensity level changes in that area. And then, rotation-invariant feature descriptors are computed.

IV. RL-EKF ALGORITHM FOR VISUAL SLAM

The proposed Reinforcement learning-based EKF algorithm is presented in detail in the following paragraphs. This algorithm is divided into two sections, reinforcement learning, and Extended Kalman Filter. The block diagram of the proposed algorithm is shown in Fig. 1.

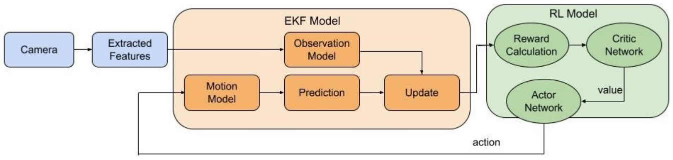

Fig. 1. The block diagram of the proposed RL-EKF feature-based visual SLAM algorithm

A. Reinforcement Learning

In this step, using the PPO or DDPG algorithm, the forward and angular velocity for the next step will be predicted in a way that reduces the uncertainty of the motion.

1) State Space:

The mean and covariance of the estimated state of the monocular camera are represented by

$$\mathrm{x} = \begin{bmatrix} \mathrm{x}_C \\ y_1 \\ y_2 \\ \vdots \end{bmatrix} \quad P = \begin{bmatrix} P_{\mathrm{xx}} & P_{\mathrm{x}y_1} & P_{\mathrm{x}y_2} & \cdots \\ P_{y_1\mathrm{x}} & P_{y_1 y_1} & P_{y_1 y_2} & \cdots \\ P_{y_2\mathrm{x}} & P_{y_2 y_1} & P_{y_2 y_2} & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix} \tag{3}$$

Where the camera state vector X_C consists of 13 parameters, including its center position r^{WC}, orientation q^{WC} (quaternion), linear velocity v^W, and angular velocity ω^C. Also, the feature states Y_i are the 3D location vectors of the measured features.

$$\mathrm{x}_C = \begin{bmatrix} \mathrm{r}^{WC} \\ \mathrm{q}^{WC} \\ \mathrm{v}^{W} \\ \omega^{C} \end{bmatrix} \tag{4}$$

The state space is represented by the mean and the unique values in the covariance matrix.

2) Action Space:

The action is represented by the linear velocity $V^W = a^W \Delta t$ and angular velocity $\Omega^C = \alpha^C \Delta t$, which are produced by the zero mean Gaussian acceleration noise $n = (a^W, \alpha^C)^T$.

3) Reward:

In this research, the reward function is designed in a way that the agent would be able to estimate the robot's position accurately in every timestep. In this regard, the goal is to minimize the difference between the robot's estimated position and its ground-truth data. In this project, in order to do that, we aim to minimize the uncertainty reduction (information gain) between timesteps, which can be achieved by computing the difference between the covariance matrix's determinant in 2 subsequent timesteps.

B. Extended Kalman Filter

1) Motion Model and Derivatives:

The camera is considered to have constant velocity and is modeled by a zero mean Gaussian acceleration noise $n = (a^W, \alpha^C)^T$. The camera's dynamic motion model is specified as below.

$$\mathrm{x}_{C_{k+1}} = \begin{bmatrix} \mathrm{r}_{\mathrm{k+1}}^{\mathrm{WC}} \\ \mathrm{q}_{\mathrm{k+1}}^{\mathrm{WC}} \\ \mathrm{v}_{\mathrm{k+1}}^{\mathrm{WC}} \\ \omega_{\mathrm{k+1}}^{\mathrm{C}} \end{bmatrix} = f_v(\mathrm{x}_{C_k}, n) = \begin{bmatrix} \mathrm{r}_{\mathrm{k}}^{\mathrm{WC}} + (\mathrm{v}_{\mathrm{k}}^{\mathrm{W}} + \mathrm{V}_{\mathrm{k}}^{\mathrm{W}})\Delta t \\ \mathrm{q}_{\mathrm{k}}^{\mathrm{WC}} \times q((\omega_{\mathrm{k}}^{\mathrm{C}} + \Omega_{\mathrm{k}}^{\mathrm{C}})\Delta t) \\ \mathrm{v}_{\mathrm{k}}^{\mathrm{W}} + \mathrm{V}_{\mathrm{k}}^{\mathrm{W}} \\ \omega_{\mathrm{k}}^{\mathrm{C}} + \Omega_{\mathrm{k}}^{\mathrm{C}} \end{bmatrix} \tag{5}$$

The motion model derivatives with respect to the state $\frac{\partial f_v}{\partial X_C}$ and the Gaussian noise $\frac{\partial f_v}{\partial n}$ are measured by Equations 4 and 5, respectively.

$$\frac{\partial f_v}{\partial \mathrm{x}_C} = \begin{pmatrix} \mathrm{I} & 0 & \Delta t\mathrm{I} & 0 \\ 0 & \frac{\partial \mathrm{q}_{\mathrm{k+1}}^{\mathrm{WC}}}{\partial \mathrm{q}_{\mathrm{k}}^{\mathrm{WC}}} & 0 & \frac{\partial \mathrm{q}_{\mathrm{k+1}}^{\mathrm{WC}}}{\partial \omega_{\mathrm{k+1}}^{\mathrm{C}}} \\ 0 & 0 & \mathrm{I} & 0 \\ 0 & 0 & 0 & \mathrm{I} \end{pmatrix} \tag{6}$$

$$\frac{\partial f_v}{\partial \mathrm{n}} = \begin{pmatrix} \Delta t\mathrm{I} & 0 \\ 0 & \frac{\partial \mathrm{q}_{\mathrm{k+1}}^{\mathrm{WC}}}{\partial \Omega_{\mathrm{k}}^{\mathrm{C}}} \\ \mathrm{I} & 0 \\ 0 & \mathrm{I} \end{pmatrix} \tag{7}$$

2) Prediction Step:

In this step, the state space vector $s_{k|k-1}$ and its uncertainty $P_{k|k-1}$ will be updated using the following equations, based on the motion model and the forward and angular velocity, which was predicted by the reinforcement Learning part of the algorithm.

$$s_{k|k-1} = f(s_{k-1|k-1}, n) \tag{8}$$

$$P_{k|k-1} = F_{k-1}P_{k-1|k-1}F^{\mathrm{T}}_{\text{k-1}}+ G_{k-1}Q_{k-1}G^{\mathrm{T}}_{\text{k-1}} \tag{9}$$

3) Feature Extraction and Matching:

For this step, features are extracted and matched using either ORBSLAM, object detection, or instance segmentation. Also, the outliers are detected using the RANSAC method.

4) Observation Model:

The observation model maps the 3D points in the world coordinate system to the image plane coordinate system. However, in order to do that, at first, the 3D points in the world coordinate system are transformed to the camera coordinate system using the equations below in terms of quaternions.

$$\mathrm{x}^c = R(\mathrm{x}^\omega - c) \tag{10}$$

$$h^{WC}(s, \mathrm{x}^\omega) = \begin{bmatrix} 0 \\ \mathrm{x}^c \end{bmatrix} = \mathrm{q} \times \begin{bmatrix} 0 \\ \mathrm{x}^\omega - c \end{bmatrix} \times q^* \tag{11}$$

And then, using the pinhole camera model, the 3D points in the camera coordinate system are mapped in the image plane coordinate system, which is done by the equation below, where f in the camera focal length.

$$h^{CI}(\mathrm{x}^c) = \mathrm{x}^I = [f\frac{x^{\mathrm{c}}_{\mathrm{i}}}{x^{\mathrm{c}}_{\mathrm{k}}}, f\frac{x^{\mathrm{c}}_{\mathrm{j}}}{x^{\mathrm{c}}_{\mathrm{k}}}] \tag{12}$$

Therefore, the whole observation model is illustrated by

$$h_{2n\times 1} = h^{CI}(h_{WC}(\mathrm{x}^\omega)) \tag{13}$$

Also, the Jacobian of the observation model with respect to the state vector is computed as below, which is used in the update step of the algorithm. The Jacobian consists of two parts, the derivatives of the observation model with respect the camera states $\frac{\partial \mathrm{h}_i}{\partial \mathrm{x}_C}$ and map features $\frac{\partial \mathrm{h}_i}{\partial \mathrm{x}_M}$.

$$\frac{\partial \mathrm{h}_i}{\partial \mathrm{x}} = (\frac{\partial \mathrm{h}}{\partial \mathrm{x}_C}\frac{\partial \mathrm{h}}{\partial \mathrm{x}_M}) \tag{14}$$

The derivatives of the observation model with respect to the camera state can be computed as below.

$$\frac{\partial \mathrm{h}}{\partial \mathrm{x}_C} = (\frac{\partial \mathrm{h}^{CI}}{\partial \mathrm{h}^{WC}}\frac{\partial \mathrm{h}^{WC}}{\partial \mathrm{x}_C}) \tag{15}$$

5) Update Step:

In this step, the range and bearing to each measurement are extracted using the RGB-D camera.Also, the Kalman gain is calculate as below.

$$K_k = P_{k|k-1}H^{\mathrm{T}}_{\mathrm{k}}(H_kP_{k|k-1}H^{\mathrm{T}}_{\mathrm{k}} + U_k)^{-1} \tag{16}$$

Using the Kalman gain, the updated state and its covariance matrix for this timestep will be computed by the equations below.

$$s_{k|k} = s_{k|k-1} + K_k(z_k - h(s_{k|k-1})) \tag{17}$$

$$P_{k|k} = (I - K_kH_k)P_{k|k-1} \tag{18}$$

where z_k is the actual observation, $h(s_{k|k-1})$ is the expected measurement, and H_k is the derivative of the observation model.

V. Refrences

A. Landmark Extraction

1) M. Kazimierczuk and J. Jozwik, “Analysis and design of class e zerocurrent-switching rectifier,” Circuits and Systems, IEEE Transactions on, vol. 37, no. 8, pp. 1000 –1009, aug 1990.
2) Hidalgo, Franco, and Thomas Bräunl. ”Evaluation of Several Feature Detectors/Extractors on Underwater Images towards vSLAM.” Sensors 20.15 (2020): 4343.
3) Mozos, Óscar Martínez, et al. ”Interest point detectors for visual slam.” Conference of the Spanish Association for Artificial Intelligence. Springer, Berlin, Heidelberg, 2007.
4) Idris, M. Y. I., et al. ”Review of feature detection techniques for simultaneous localization and mapping and system on chip approach.” Information Technology Journal 8.3 (2009): 250-262.
5) Li, Guangqiang, Lei Yu, and Shumin Fei. ”A deep-learning real-time visual SLAM system based on multi-task feature extraction network and self-supervised feature points.” Measurement 168 (2021): 108403.
6) Rublee, Ethan, et al. ”ORB: An efficient alternative to SIFT or SURF.” 2011 International conference on computer vision. Ieee, 2011.
7) Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. ”Surf: Speeded up robust features.” European conference on computer vision. Springer, Berlin, Heidelberg, 2006.
8) Calonder, Michael, et al. ”Brief: Binary robust independent elementary features.” European conference on computer vision. Springer, Berlin, Heidelberg, 2010.

B. (EKF) Visual SLAM

1) Se, Stephen, David Lowe, and Jim Little. ”Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks.” The international Journal of robotics Research 21.8 (2002): 735-758.
2) Casarrubias-Vargas, Heriberto, Alberto Petrilli-Barceló, and Eduardo Bayro-Corrochano. ”EKF-SLAM and machine learning techniques for visual robot navigation.” 2010 20th International Conference on Pattern Recognition. IEEE, 2010.
3) Yuan, Wang, Zhijun Li, and Chun-Yi Su. ”RGB-D sensor-based visual SLAM for localization and navigation of indoor mobile robot.” 2016 International Conference on Advanced Robotics and Mechatronics (ICARM). IEEE, 2016.
4) Siciliano, Bruno, and Oussama Khatib, eds. Springer handbook of robotics. springer, 2016.
5) Murphy, Timothy Charles. Examining the Effects of Key Point Detector and Descriptors on 3D Visual SLAM. Diss. Ohio University, 2016.

6) Civera, Javier, Andrew J. Davison, and JM Martinez Montiel. ”Inverse depth parametrization for monocular SLAM.” IEEE transactions on robotics 24.5 (2008): 932-945.
7) Ceriani, Simone, et al. ”On feature parameterization for ekf-based monocular slam.” IFAC Proceedings Volumes 44.1 (2011): 6829-6834.
8) Thrun, Sebastian. ”Probabilistic robotics.” Communications of the ACM 45.3 (2002): 52-57.
9) Yousif, Khalid, Alireza Bab-Hadiashar, and Reza Hoseinnezhad. ”An overview to visual odometry and visual SLAM: Applications to mobile robotics.” Intelligent Industrial Systems 1.4 (2015): 289-311.
10) Chaplot, Devendra Singh, Emilio Parisotto, and Ruslan Salakhutdinov. ”Active neural localization.” arXiv preprint arXiv:1801.08214 (2018).
11) Chaplot, Devendra Singh, et al. ”Learning to explore using active neural slam.” arXiv preprint arXiv:2004.05155 (2020).
12) Mehralian, Mohammad Amin, and Mohsen Soryani. ”EKFPnP: extended Kalman filter for camera pose estimation in a sequence of images.” IET Image Processing (2020).
13) Civera, Javier, Andrew J. Davison, and José María Martínez Montiel. Structure from motion using the extended Kalman filter. Vol. 75. Springer Science Business Media, 2011.

C. *Reinfrocement Learning*

1) Shukla, Ayush. Reinforcement Learning based Active Localization for precise manipulation. Diss. Georgia Institute of Technology, 2020.

D. *Habitat Simulator and Datasets*

1) Xia, Fei, et al. ”Gibson env: Real-world perception for embodied agents.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
2) Savva, Manolis, et al. ”Habitat: A platform for embodied ai research.” Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.
3) Chang, Angel, et al. ”Matterport3d: Learning from rgb-d data in indoor environments.” arXiv preprint arXiv:1709.06158 (2017).
4) Straub, Julian, et al. ”The Replica dataset: A digital replica of indoor spaces.” arXiv preprint arXiv:1906.05797 (2019).

E. *Object Detection and Semantic Segmentation*

1) Chaplot, Devendra Singh, et al. ”Semantic curiosity for active visual learning.” European Conference on Computer Vision. Springer, Cham, 2020.
2) Hou, Ji, Angela Dai, and Matthias Nießner. ”3d-sis: 3d semantic instance segmentation of rgb-d scans.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.