Simultaneous Localization and Mapping of a Mobile Robot With Stereo Camera Using ORB features

Submitted: 2nd November 2022; accepted: 18th January 2023

Younès Raoui, Mohammed Amraoui

DOI: 10.14313/JAMRIS/2-2024/14

Abstract:
Simultaneous Localization and Mapping (SLAM) is applied to robots for accurate navigation. The stereo cameras are suitable for visual SLAM as they can give the depth of the visual landmarks and more precise estimations of the robot's pose. In this paper, we present a survey of SLAM methods, either Bayesian or bio-inspired. Then we present a new method of SLAM, which we call stereo Extended Kalman Filter, improving the matching by computing the innovation matrices from the left and the right images. The landmarks are computed from Oriented FAST and Rotated BRIEF (ORB) features for detecting salient points and their descriptors. The covariance matrices of the state and the robot's map are reduced during the robot's motion. Experiments are done on the raw images of the Kitti dataset.

Keywords: *Simultaneous localization and mapping, stereo cameras, extended Kalman filter, mobile robots, navigation*

1. Introduction

Simultaneous Localization and Mapping (SLAM) is an essential task for robot navigation. It allows the robot to reach its goal without errors because it is suitable for planning an optimal path and avoiding obstacles. The robot uses several sensors to construct maps, such as cameras, LIDAR arrays, or IMUs. As maps are computed when the robot moves, localization is vital for mapping. Several new mapping methods have recently appeared to allow the robot to be autonomous in indoor environments, allowing robots to do tasks such as cleaning houses. Also, outdoor environments must be mapped for tasks such as autonomous driving.

The robot SLAM includes several techniques, which can be divided into Bayesian and bio-inspired methods using neural networks and deep learning. The Bayesian approach constructs maps by taking data from sensors and building representations of the environment using landmarks, point clouds, or graphs. The bio-inspired techniques are inspired by neuroscience, especially the place cells and the grid cells to build representations. Another distinction between SLAM maps is that they can be metrical or topological. The metrical maps are based on the extraction and tracking of landmarks computed using the SLAM methods.

The topological maps construct graphs representing places and then link those places with edges. The edges in topological maps are suitable for planning as they show the cost of navigation from one place to another by avoiding obstacles. More attention is given to each technique in the following paragraphs.

The Bayesian mapping methods are divided into those that use landmarks and those that use objects. Both optimize the robot's pose and the map regarding the motion commands and the observations. Mapping with landmarks is generally based on creating sparse representations of the environment with the vision or laser scans. Mapping with objects uses deep learning techniques to detect the objects; these deep learning detections require increased GPU performances but are more accurate. The data association in such scenarios is accomplished using semantic labels to associate the 3D objects in the map with the new measures. Moreover, topological maps with objects use deep learning methods for object detection and scene segmentation. They use the concept of place, a node in a graph representing, for instance, an office and a kitchen, and the relation between the places' edges. Each place could be a class obtained with scene segmentation with Convolutional Neural Networks or a set of objects.

In contrast, bio-inspired methods rely upon the domain of navigation of the mammal brains. Grid cells and the place cells in the mammalian entorhinal cortex and the hippocampus are responsible for localization and mapping. These cells provide machinery for the indexing of poses and places through path integration and perceptions. The mammal can construct a topological map that allows it to navigate accurately to its goal. Bio-inspired methods aim to replicate this process in robotic SLAM applications.

This paper aims first to make a short survey of SLAM methods and then to present a new method of SLAM with stereo cameras using ORB features. The new method extends the Monocular SLAM but uses the ORB features suitable for SLAM due to their robustness when viewpoints and scale change. This new method improves the active search algorithms used in Monocular SLAM by considering the right and left images measurements to increase the number of matchings.

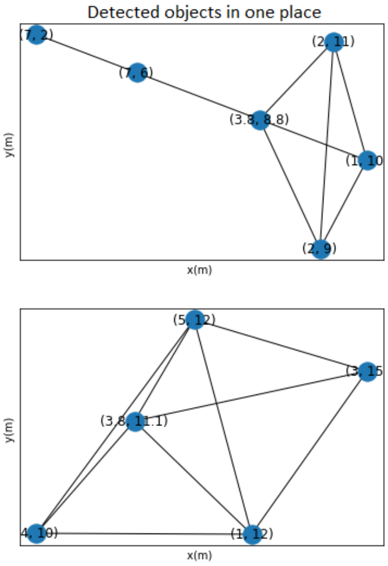

Figure 1. The creation of two places using objects detected with Convolutional Neural Networks by considering each object as a vertex in a graph and two places are connected with an edge

The paper is organized as follows: First, we present a survey of visual SLAM methods; second, our methodology, which consists of fusing information from the stereo camera to increase the number of correct matchings in filtering the observations, third, we adress the experiments and results; finally, we present the conclusion.

2. A Survey of SLAM Techniques

Visual SLAM using landmarks

The paper [18] presents a method of localization and mapping with Radio Frequency Identifiers, which uses the Blackwellized localization technique based on particle filters. It extends the metrical mapping with a topological map for trolley navigation in a supermarket. The paper [4] presents a survey of the SLAM methods during the last decade. The seminal work of visual SLAM is [6], which used Extended Kalman Filter (EKF) with a monocular camera. The work [15] presents a method of visual SLAM that initializes the landmark using their inverse depth. This measurement changes when the camera moves, and thus its covariance is reduced. When a new landmark is created, the number of detected landmarks is low, and its inverse depth is also initialized. The ORB SLAM 2 and 3 uses bundle adjustment instead of EKF to update the current pose and all the poses from the start position [16], [5]. It describes the scene with the visual bag of words DBow2 library [10] , which are used for place recognition as well.

2.1. Brain Inspired SLAM

The work in [2] is significant as it builds a system of localization and mapping using the grid and place cells knowledge. It uses a continuous attractor network to implement a neural field to model the robot's path integration.

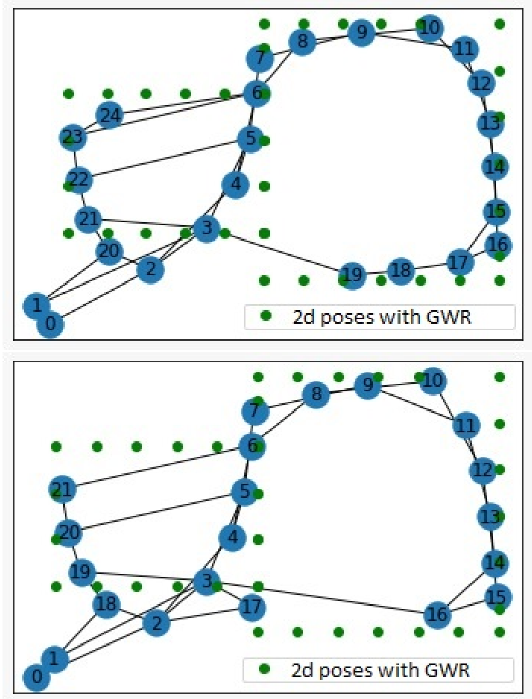

Figure 2. 2 maps created with the Growing When Required Network by clustering the places by computing distances between the current one and the others based on the activity of the neurons

It builds a Hebbian neural network to model the place recognition of the robot. A map is built to show in the world frame the activity of the pose cell network. This map is semi-topological, and it connects the nodes with edges representing the direction of the motion of the robot's head. A path planning Dijkstra algorithm is implemented as well in the RatSLAM. An extension of RatSLAM called NeuroSLAM is developed to deal with 3 dimensions experience maps, including the XYZ world frame coordinates. In this work, the depth is also coded in the pose cell network [28]. In [19], a new brain-inspired SLAM system that extends the NeuroSLAM and exploits the semantics of scenes is developed. It trains a continuous attractor network by stimulating it with visual objects. Then, it constructs 2D/3D experience maps and relax them.

2.2. SLAM with Objects

[26] built a SLAM system with dynamic objects at different semantic levels. They proposed a missed detection compensation algorithm based on the speed invariance in adjacent frames. [27] proposed a real-time and robust visual SLAM system based on ORB-SLAM2 for a dynamic environment. They incorporated deep learning object detection for preprocessing data relied on dynamic targets or static objects. They reduced the tracking errors and enhanced the accuracy of the computation of the pose and the map. [7] worked with semantic and geometric uncertainty inherent in object detection methods, modeled the non-linear data association with objects, and developed a max-mixture type model that accounts for multiple data association hypotheses for object detection

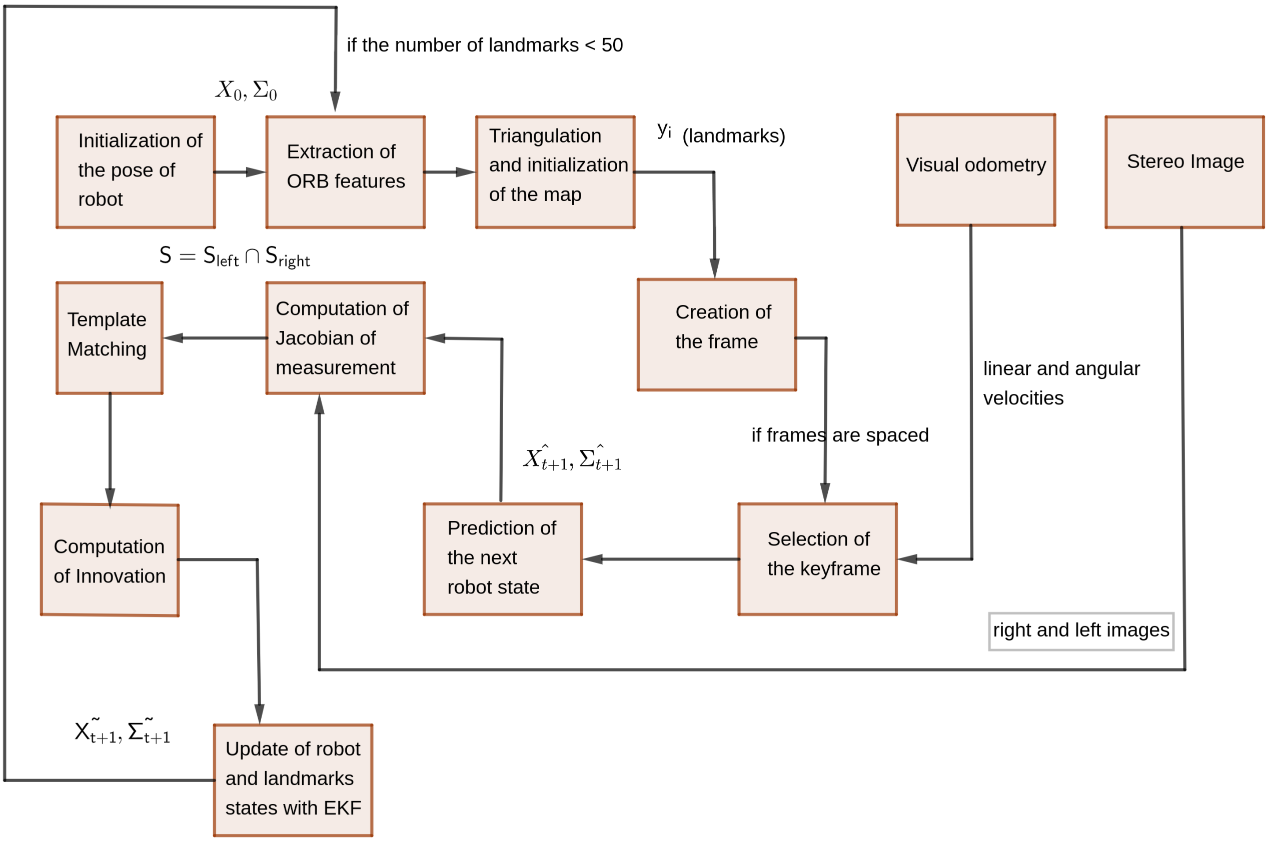
if the number of landmarks < 50
X_0, Σ_0
Initialization of the pose of robot
Extraction of ORB features
Triangulation and initialization of the map
y_i (landmarks)
Visual odometry
Stereo Image
$S = S_{left} \cap S_{right}$
Template Matching
Computation of Jacobian of measurement
Creation of the frame
linear and angular velocities
if frames are spaced
$\hat{X}_{t+1}, \hat{\Sigma}_{t+1}$
Computation of Innovation
Prediction of the next robot state
Selection of the keyframe
right and left images
$\tilde{X}_{t+1}, \tilde{\Sigma}_{t+1}$
Update of robot and landmarks states with EKF

Figure 3. Overview of our method

with Mobile Net Single-Shot multibox Detection (SSD). [8] solved the semantic SLAM problem using non-parametric belief propagation. [22] formulated lidar inertial odometry atop a factor graph by marginalizing old lidar scans to a global map.

2.3. Place Recognition

The survey in [14] [13] presented several methods for Visual Place Recognition (VPR) and how it is shaped by the recent advances in deep learning. Then the survey shows how metric-learned techniques are used to develop such systems in the presence of occlusions and distortions in the images. They concluded that the future trends in image classification, object detection, and methods related to learning objects such as buildings would affect long-term place recognition. In recent research, deep learning methods have become useful in VPR, such as in [24] where they boosted the performance of learned features with geometric transformations based on reasonable domain assumptions about navigation on a ground plane. Besides, detecting loop closures with the VPR approach is very important as it allows correction of the robot's trajectory on a topological map. In ORB-SLAM 2 and 3 [5, 16], they detect candidates of loops by extracting and comparing a visual bag of words ORB features from the current keyframe and those not directly connected to this keyframe and which are connected in the Covisibility graph. [25] developed a new deep learning technique to detect loop closures using hierarchical retrieval of frames from video streams collected by a mobile robot.

In [17], a self-supervised approach using deep learning is proposed to allow robot learning from examples of loops that used GPS-aided navigation. The robot learned a distance metric for visual descriptors using Convolutional Neural Networks for place recognition. They used a corset method to detect loops for retrieving frames [25] . There are several methods to detect places and create maps. Another application of VPR systems is using objects instead of visual features. In [9], objects are detected from Kinect 3D data used to build a topological graph of objects (see Fig. 1). Then from this graph, a topological map is created where places are detected based on the distance between objects. A pair-wise matching score between the current and previous places is computed to decide if a place is familiar to detect the loop. This pair-wise matching score depends on the environment. Image-quality high-resolution 3D point clouds can also be used to identify places by describing scenes with DBoW features and then applying Perspective-n-Point (PnP) to detect places [23]. Finally, places can also be detected using semantic segmentation to create sufle maps from registered frames by using Conditional Random Fields or semantic labeling to correlate objects to scenes extracted with convolutional neural networks [1].

2.4. Brain Based Navigation

The navigation using bio-inspired models can be done for mapping or localization. One of the seminal works in modeling the associative learning in place cells and grid cells is in [20].

An associative neural network models the positioning of a mammal with place cells for place recognition or grid cells for localization at different scales in the space.

Another model for explaining connections between place cells and grid cells is developed in [12]. This one has inputs and outputs from place cell activities. By learning weights of connections to the hidden layer representing the grid cells, triangular patterns are generated. Inhibitory connections are used between the neurons of the hidden layers, which makes the model recurrent as well.

The navigation is not only relied on the place and grid cells but also to object vector cells used by the brain to memorize the places that visited an animal by taking as landmarks the objects regardless to their type [11]. It has been shown that the object vector cells spike when the animal is near an object. Each object relies on one or several neurons. The neurons' firing field depends on the distance and the vector between the animal and the detected object. Those cells create a memory of the animal able to know when a place is familiar.

The Growing When Required networks are also used in [29] to model the place cells activities. This auto-associative network can create maps of the robot for planning in a topological graph where nodes are the places, and the edges are the path between places (Fig. 2). The cost of each path depends on the existence of an obstacle. Thus, a direct and forward model of the robot motion is developed with dynamic neural fields, which take as stimuli the camera data and predicts the actions that the robot should take [29]. The slowness of the variation of the activities of the place cells and head direction cells is modeled with Slow Feature Analysis.

The modeling of the activity of grid cells in complex environments has been performed with deep reinforcement learning networks in [3]. Patterns similar to the experiments are obtained by training a deep neural network that learns from the place cells' activities and head direction cells. Therefore, the artificial agent can localize itself in an unfamiliar environment using the model.

3. Presentation of the StereoSLAM with Fusion of Innovation Jacobians

3.1. Overview and Contribution

We present in this paper a new method of visual Simultaneous Localization and Mapping (Fig. 3) using stereo cameras which provides the depth of the 3d camera points. The problem of SLAM is tackled with the filtering approach using Extended Kalman Filter to estimate the pose of the robot and the maps. We start by initializing our system with a set of landmarks computed with triangulation of ORB feature points (Fig. 4). The state of the camera (Eq. 1) is updated with the perception of those landmarks. As the robot acquires stereo images, it has as an observation of the right and the left images.

When using only the right or the left images, we noticed some steps where the robot could not update the pose with EKF caused by the lack of correct matchings or because the jacobian of the innovation is very high. When it is the case, the active search region in the new frame cannot be processed because it depends on the eigenvalues of the covariance of the innovation. Therefore, we compute the innovation matrices for the right and left images and do matching against the ORB feature points using template matchings. Then we fuse the two measures by discarding the redundant features. The steps characterizing our system are as following (see algorithm):

```
Result: Algorithm of Stereo EKF SLAM
matched_landmarks={}
  X=null
  for i ≤ len(way_points) do
      if map ={} then
          initialize_map()
            initialize_state()
      end
      else
          if len(matched_landmarks ≤ 50 then
              initialize_new_landmarks()
          end
          [$\tilde{X}_t, \hat{\Sigma}_t$]=prediction(slam, robot_model)
            compute_covariance_of_new_landmarks()
            for j ≤ len(all_landmarks) do
              if landmark[j] is observable then
                  $h_{tr\ j}$=inverse_stenope_model
                    (landmark[j], right)
                    $h_{tl\ j}$=inverse_stenope_model
                    (landmark[j], left)
                    $H_{il}$=observationJacobian($h_{tr\ j}$)
                    $H_{ir}$=observationJacobian($h_{tr\ j}$)
              end
          end
          $S_r$=compute_innovation_matrix_right()
            $S_l$=compute_innovation_matrix_left()
            $z_t$ = matching($S_r, S_l$)
            [$\tilde{X}_t, \tilde{\Sigma}_t$] = update($\hat{X}_t, \hat{\Sigma}_t, \hat{z}_t, h_t$)
      end
end
```

1) **Extraction of ORB features:** We choose ORB features for the scene understanding as they are invariant to rotation and multi-scale. Thus when we use them, the robot could update the landmarks of the map points (landmarks). They are extracted with a rate of 0.03 second per image, which is performant comparing to SURF 0.04 second or SIFT 0.13 second. The ORB descriptor uses a rotated version of BRIEF according to the orientation of the key points [21].

2) **Initialization the map:** The map is initialized with 50 landmarks computed by triangulating ORB features in the right and left images. The initial poses of the landmarks are computed at the reference frame. The initial pose of the camera is taken at the reference position also.

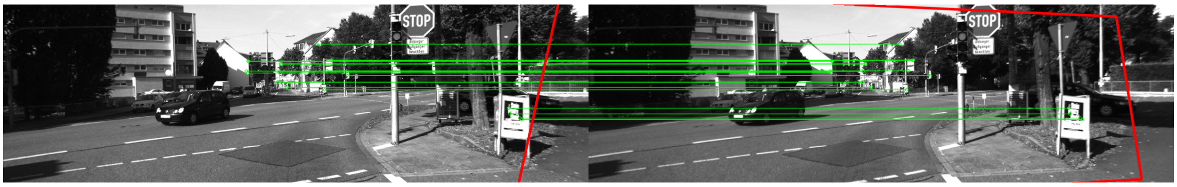

Figure 4. Matching of ORB features between two left and right images by taking 20 matchings using the correspondance of keypoints inside the red rectangle plane

3) **Extraction of frames and keyframes:** We extract a frame composed of the robot's pose at each step, including 3d coordinates and the quaternion, and the angular and linear velocities. It also includes the ORB feature points and their BRIEF descriptors. We create a Keyframe if the robot passes four frames.
4) **Computation of the innovation:** We project the 3D map points in the right and the left cameras, and we compute the two innovation matrices. Then we do matchings in the left and the right images using the two innovation matrices which are of size (2,2).
5) **Filtering with EKF:** The filtering is done for each keyframe by applying the Extended Kalman Filter (EKF) . The update is done by projecting the landmarks of the map in the right and left image planes. The matching points are used for the update.

3.2. State Vector of the Stereo Camera

Let's have the state of the camera fixed on the robot given with the following vector (Eqs. 1, 2):

$$X_t = \begin{bmatrix} r_{wc} & q_i & V_i & \omega_i \end{bmatrix} \tag{1}$$

Where r_{wc} is the 3d pose of the camera in the absolute frame, q_i is the orientation quaternion, V_i is the linear velocity, and ω_i is the angular velocity.

The covariance of the pose of the camera is a 13*13 matrix represented by:

$$\begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \cdots & \sigma_{x\omega} \\ \sigma_{yx} & \sigma_{yy} & \cdots & \sigma_{y\omega} \\ \vdots & & & \\ \sigma_{\omega x} & \sigma_{\omega y} & \cdots & \sigma_{\omega\omega} \end{bmatrix} \tag{2}$$

3.3. The Representation of the Visual Feature

We represent each visual feature as a Cartesian vector in the world $[y_i = X_i, Y_i, Z_i]$ where Z_i is computed by the triangulation of ORB features from the left and the right image planes using the baseline of the stereo camera. We start by initializing the map by several features obtained by a stereo camera. We use ORB features as salient features to describe the first frame. To obtain the features in the camera frame, we apply the triangulation Equation (3):

$$x_{3D} = \begin{bmatrix} \frac{b.(u_r - u_0)}{d} & \frac{b.(v_r - v_0)}{d} & \frac{f.b}{d} \end{bmatrix}^T \tag{3}$$

Where b is the baseline expressed in meter, u_0 and v_0 are the coordinates of the optical center, f is the focal distance expressed in pixels, and u_r and v_r are the coordinates of the matched feature in the right image.

The depth d is computed using the triangulation of matching feature points by considering the epipolar constaint. In the world frame, the landmark (map point) is given with the following equation:

$$y_i = R_{cw} * x_{3D} + t_{cw} \tag{4}$$

Where R_{cw} and t_{cw} are the rotation and the translation matrices from the robot to the world frame given with the visual odometry module of the robot.

The motion model of the camera

We obtain the linear and angular velocities from visual odometry using viso2_ros node in the Robotic and Operation System (ROS). We modified the motion model in [6] as the velocity is not constant (Eq. 5). We start by initializing the first pose of the camera; then, we compute the new 3d pose by multiplying the previous pose by the rotation matrix R_{cw} and adding the product of the velocity and the time slot between two successive frames.

$$[x_t, y_t, z_t] = R_{cw} * \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \end{bmatrix} + V_{t-1} * dt \tag{5}$$

V_t has 3 components v_x, v_y and v_z.dt is the time difference from the t-1 to t

We convert the rotation quaternion of the camera to Euler angles, ψ, θ, and ϕ.

We update those angles using the angular velocity vector ω (Eqs. 6, 7, 8):

$$\psi_t = \psi_{t-1} + \omega_x * d_{t-1} \tag{6}$$

$$\theta_t = \theta_{t-1} + \omega_y * d_{t-1} \tag{7}$$

$$\phi_t = \phi_{t-1} + \omega_z * d_{t-1} \tag{8}$$

Where $\omega = [\omega_x, \omega_y, \omega_z]$ is the angular velocity of the camera.

Then we express ψ_t, θ_t, ϕ_t in terms of the quaternion.

The next camera state is expressed with the vector of dimension (13, 1) (Eq. 9):

$$\begin{bmatrix} x_t & y_t & z_t & quaternion_t & V_t & \omega_t \end{bmatrix} \tag{9}$$

Observations of the landmarks

The camera captures the image, and the robot observes the nearest landmarks on the map. By matching map points to image points, the robot can correct its pose. Then it should project them using the inverse pinhole model to the new keyframe.

Let us have the pose of the landmark y_i in the world frame. The pose of this feature in the camera frame is given with its observation:

$$h_L^R = R^{RW}(y_i^W - r^W) \tag{10}$$

Where R^{RW} is the inverse of the rotation matrix given by transforming the quaternion of the robot state to a rotation matrix, and r^W is the pose of the camera in the world frame.

To get coordinates of the predicted feature, we multiply it by the intrinsic parameters of the right and the left cameras:

For the left camera, the observation is given with (Eq. 11).

$$h_i = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_{l0} - fkl_u \frac{h_{Lx}^R}{h_{Lz}^R} \\ v_{l0} - fkl_v \frac{h_{Ly}^R}{h_{Lz}^R} \end{bmatrix} \tag{11}$$

Where u_{l0}, v_{l0}, fkl_u and fkl_v are the standard camera calibration parameters.

The same equation is for the right camera expects that we use its specific intrinsic parameters.

3.4. Active Search

We project the landmarks in the left and the right images to make data associations between the landmarks and the detected ORB features associated with them. We project on the left and the right image planes to compute the left and right innovations matrices. Then we look for the best matchings using the template matching algorithm for the two images. We consider the two planes because when we use only the left or the right image, it appears that in some places, we do not have a lot of associations or no one at all. Thus, the update is not accurate. With our method, at each step, we have good association.

Estimation of the camera pose

The Extended Kalman Filter applied to visual SLAM with stereo cameras is similar to one of the monocular cameras in [6] and [15] in such a way that it uses the active search to match the predicted features and the measured ones. Indeed, To determine the area in which we apply template matching, we should compute the Jacobian matrix S_i given with (Eq. 12):

$$S_i = \frac{\delta u_{di}}{\delta x_v} P_{xx} \frac{\delta u_{di}}{\delta x_v}^T + \frac{\delta u_{di}}{\delta x_v} P_{xy_i} \frac{\delta u_{di}}{\delta y_i}^T + \frac{\delta u_{di}}{\delta y_i} P_{y_i x} \frac{\delta u_{di}}{\delta x_v}^T + \frac{\delta u_{di}}{\delta y_i} P_{y_i y_i} \frac{\delta u_{di}}{\delta y_i}^T + R \tag{12}$$

here x_v is the pose of the camera, u_{di} is the distorted feature point in the image, and y_i is the feature in the world frame. Also, R is the noise of the observation of pixels.

Also, P_{xx} is the covariance of the robot pose of size (13*13). P_{xy_i} is the covariance between the camera pose and the feature a 13*6 matrix, and $P_{y_i x}$ is the transpose of P_{xy_i}, and $\mathrm{P}_{y_i y_i}$ is the covariance of the features a 3*3 matrix.

In order to compute the error on the feature, we should compute the covariance of each initialized feature (Eq. 13): In the context of stereo camera, the computation of $\frac{\delta y_i}{\delta h}$ is done with the following equation because we have 3 components of the image point:

$$\frac{\delta y_i}{\delta h} = \begin{bmatrix} \frac{b * x_l}{x_l - x_r}^2 & 0 & 0 \\ -b \frac{b * y_l}{x_l - x_r} & \frac{b}{x_l - x_r} & 0 \\ -b(\frac{f}{x_l - x_r})^2 & 0 & 0 \end{bmatrix} \tag{13}$$

Where (x_l, y_l) and (x_r, y_r) are the image coordinates in the left image and its corresponding in the right image, and b is the baseline, and f is the focal length.

The jacobian S (12) is computed for the two planes and the matching is performed using the left and right jacobians. Indeed, some points have the jacobian very high, so the uncertainty ellipse around the measured ORB feature point is not suitable. That is why we should use a lot of matchings.

Estimation of the camera pose and its covariance with EKF SLAM

We use the Extended Kalman Filter to compute the new estimated pose of a robot, for that we compute the Kalman gain using the equation (Eq. 14):

$$K = H * PH^T + (H * P * H^T + R)^{-1} \tag{14}$$

Where:

H is the jacobian of the measures computed using the model of observation of the stereo camera. We compute H for the right and the left cameras by measuring each landmark. It is observable if it is projected on the image plane. The jacobians are computed by differentiating the h function (see Eq. 10) by the pose of the camera and its quaternion. We compute the updated pose of the camera using the equation (Eq. 15):

$$\tilde{X_t} = \hat{X_t} + K_t * (z_t - h(\hat{X_t})) \tag{15}$$

Where $\tilde{X_t}$ is the predicted robot state with the motion model, z_t is the measured matched map points. Moreover we compute the updated covariance using this formula (Eq. 16):

$$P_{t+1}^{\tilde{}} = (I - K_{t+1}.H_{t+1})P_{t+1}^{\hat{}} \tag{16}$$

Where $P_{t+1}^{\hat{}}$ is the predicted covariance of the camera and I is an identity matrix of size (13+numbre of features)*(13+numbre of features).

4. Results and Discussion

Experiments

We have implemented our system using the Python language on ROS melodic distribution. We used the Kitti dataset (2011_09_26_drive_0017) for the tests. The odometry data is obtained with the visual odometry node (viso_2 ros), which we run in offline to obtain the linear and angular velocities.

Table 1. The intrinsic parameters of the stereo camera (left and right planes) have the same values

f_x	$7.21e^{-4}$ m
f_y	$7.21e^{-4}$ m
c_x	1242 pixels
c_y	375 pixels
d_x	10^{e-6} m
d_y	10^{e-6} m
baseline	0.06216 m

Table 2. Noise of velocities

Noise of linear velocity	6m/s
Noise of angular velocity	1 rad/s

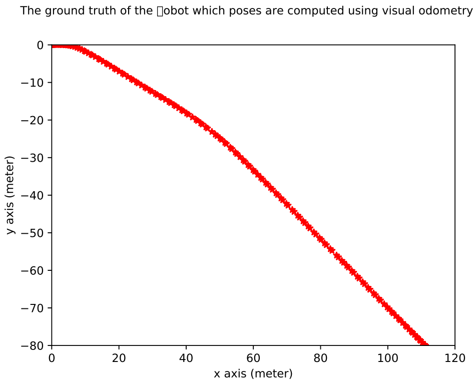

Figure 5. The trajectory obtained with visual odometry with the viso2ros package of ROS

The tracking method is implemented in a Python thread to allow parallel programming such as in [16].

We used the calibration data available in the Kitti dataset (table (1).

Our system allows us to update the camera pose and the landmarks with the EKF. The choice of the noises on the image pixels and the robot poses is as follows: 5 pîxels for computing covariance of the pixel measurements R_i, 6m/s for the linear velocity, and 1rad/s for the angular velocity.

4.1. The Trajectory of the Camera Using Visual Odometry

We used viso_2ros package to compute the trajectory of the kitti car using the mono_odometer node. We obtain the following trajectory (Fig. 5):

We add some noise to this trajectory to apply the prediction step of the Extended Kalman Filtering (Fig. 2). the red line represents the true trajectory of the robot which is used to evalueate the algorithm of position computation. Using the prediction equations, we obtain the following figure where we remark that the covariance of the position increases, which is normal as the robot does not correct its position regarding the ground truth.

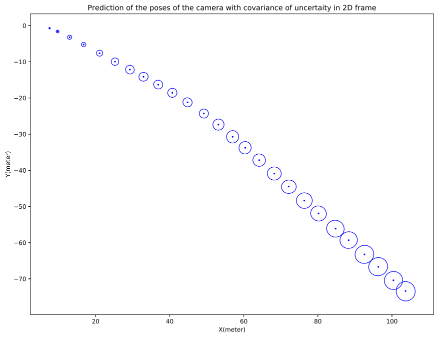

Figure 6. Predicted positions and covariances of the camera

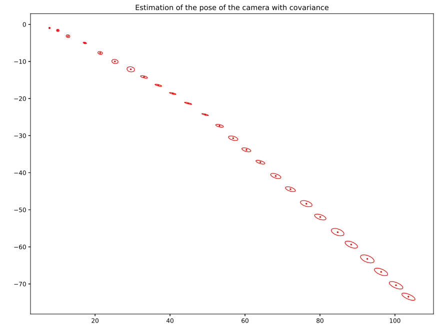

Figure 7. The estimation of the pose of the robot in 2D. The ellipses of the covariance are reduced inside the trajectory of the robot due to the application of EKF

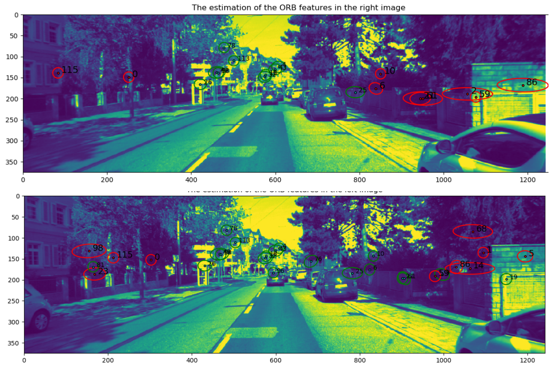

Figure 8. The projection of the landmarks of the map on the right and the left images in the time T_1. The red ellipses represent the covariances of the near landmarks observed in the new keyframe.

4.2. Update of the Pose of Robot

We update the state of the robot (Fig. 7) by filtering noise accumulated on the linear and angular velocities, which depends on the location of the camera and its quaternion. The following figure shows the estimation of the robot poses with covariances reduced comparing to the predicted case (Fig. 6). The covariance of the error on the estimation of the robot pose is reduced after the robot is moving.

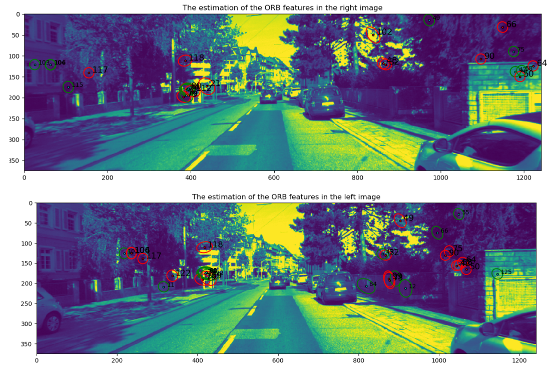
The estimation of the ORB features in the right image
The estimation of the ORB features in the left image

Figure 9. The projection of the landmarks of the map on the right and the left images in the time T_2. The red ellipses represent the covariances of the near landmarks observed in the new keyframe.

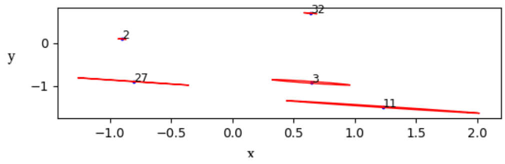

Figure 10. Detected landmarks using the projection on the camera model in the plane XY

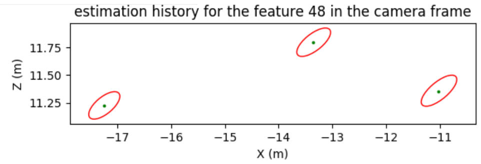

Figure 11. The tracking of the landmark 48 with the ellipses of the covariance in the plane XZ (Z is the depth)

4.3. Update and Management of Map

To update the map, we use the correction step of the EKF but after doing an active search of the correct matches to reobserve the map points and reduce their uncertainty. The matching threshold of the template matching is set to 0.8. The patch of the template matching is set to 20 pixels. We consider the undistorted ORB feature points in our implementation.

The figures (Figs. 8, 9) show the tracking of the ORB features with EKF.

Moreover, the computed landmarks at time t show their XY poses with uncertainty (Fig. 10). This latest is reduced when the camera moves because of the application of the Kalman gain. We also show the evolution of the landmark of number 48 (Fig. 11). In this figure, we represent the XZ coordinates in the image plane to view the depth of the features. This latest should be lower than a threshold to accept it. This threshold is set in our experiment to $40 * baseline$.

4.4. Discussion

We remark in the Figure 7 that the ellipse of covariances is reduced because of the application of the update of the extended Kalman filter. When we compare to the Figure 6, the ellipses are smaller. When we increase the number of extracted ORB features, the result is better, but the computation time is higher because of the complexity of the template matching. Also, if the number of observable landmarks is lower than a threshold at each keyframe, we should add new features to the map. We set the threshold to 50 features for our experiment. otherwize we keep only 50 observations from the camera which are taken randomly.

5. Conclusion

In this paper, we presented a new method of stereo SLAM using the ORB feature points. We improved the jacobian matrix of the image observation of the update of the EKF to include matching from the left and the right images using two jacobians of the variation of the measures of the ORB features. This optimize the filtering with EKF. We proved that our method reduces the error's covariance associated with the state of the robot and the map. We mentioned that stereo cameras with ORB features are convenient for visual SLAM as we could get the depth by triangulation without inverse depth parametrization. From perspective, we could use visual objects instead of landmarks to get the observations and use a neural network learning algorithm for creating the metrical map such as Growing When Required Networks. We could also develop a SLAM based on the poses of landmarks using a competitive neural network.

AUTHORS

Younès Raoui – Physics Department, Laboratory Conception and Systems, Faculty of Sciences, Mohammed V University in Rabat, 4, Avenue Ibn Battouta, BP 1014, Rabat, Morocco, e-mail: y.raoui@um5r.ac.ma.

Mohammed Amraoui – Computer Sciences Department, Intelligent Processing and Security Systems Team, Faculty of Sciences, Mohammed V University in Rabat, 4, Avenue Ibn Battouta, BP 1014, Rabat, Morocco, e-mail: m.amraoui@um5r.ac.ma.

References

[1] Ambrus, R., Claici, S., Wendt, A.: Automatic room segmentation from unstructured 3-d data of indoor environments. IEEE Robotics and Automation Letters 2, 749–756 (2017)

[2] Ball, D., Heath, S., Wiles, J., Wyeth, G., Corke, P., Milford, M.: Openratslam: an open source brain-based slam system. Autonomous Robots 34, 1–28 (04 2013). doi: 10.1007/s10514-012-9317-9

[3] Banino, A., Barry, C., Uria, B., Blundell, C., Lillicrap, T., Mirowski, P., Pritzel, A., Chadwick, M.J., Degris, T., Modayil, J., Wayne, G., Soyer, H., Viola, F., Zhang, B., Goroshin, R., Rabinowitz, N.C., Pascanu, R., Beattie, C., Petersen, S., Sadik, A., Gaffney, S., King, H., Kavukcuoglu, K., Hassabis, D., Hadsell, R., Kumaran, D.: Vector-based navigation using grid-like representations in artificial agents. Nature 557, 429–433 (2018)

[4] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I.D., Leonard, J.J.: Simultaneous localization and mapping: Present, future, and the robust-perception age. CoRR abs/1606.05830 (2016), http://arxiv.org/abs/1606.05830

[5] Campos, C., Elvira, R., Rodríguez, J.J.G., Montiel, J.M.M., Tardós, J.D.: Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam (2020)

[6] Davison, A., Reid, I., Molton, N., Stasse, O.: Monoslam: Real-time single camera slam. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 1052–1067 (2007)

[7] Doherty, K., Baxter, D., Schneeweiss, E., Leonard, J.: Probabilistic data association via mixture models for robust semantic slam. 2020 IEEE International Conference on Robotics and Automation (ICRA) pp. 1098–1104 (2020)

[8] Doherty, K., Fourie, D., Leonard, J.: Multimodal semantic slam with probabilistic data association. 2019 International Conference on Robotics and Automation (ICRA) pp. 2419–2425 (2019)

[9] Finman, R., Paull, L., Leonard, J.: Toward object-based place recognition in dense rgb-d maps. 2015 IEEE International Conference on Robotics and Automation (ICRA) (2015)

[10] Gálvez-López, D., Tardós, J.D.: Bags of binary words for fast place recognition in image sequences. IEEE Transactions on Robotics 28, 1188–1197 (2012)

[11] HÃ¸ydal, Ã., SkytÃ¸en, E., Andersson, S., Moser, M.B., Moser, E.: Object-vector coding in the medial entorhinal cortex. Nature 568, 1–8 (04 2019). doi: 10.1038/s41586-019-1077-7

[12] Kiggundu, A., Weber, C., Wermter, S.: A compressing auto-encoder as a developmental model of grid cells (02 2017)

[13] Lowry, S.M., Sünderhauf, N., Newman, P., Leonard, J.J., Cox, D.D., Corke, P.I., Milford, M.J.: Visual place recognition: A survey. IEEE Trans. Robotics 32(1), 1–19 (2016). doi: 10.1109/TRO.2015.2496823.

[14] Masone, C., Caputo, B.: A survey on deep visual place recognition. IEEE Access 9, 19516–19547 (2021). doi: 10.1109/ACCESS.2021.3054937,

[15] Montiel, J., Civera, J., Davison, A.: Unified inverse depth parametrization for monocular slam. In: Robotics: Science and Systems (2006)

[16] Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics 33, 1255–1262 (2017)

[17] Pillai, S., Leonard, J.J.: Self-supervised visual place recognition learning in mobile robots. CoRR abs/1905.04453 (2019), http://arxiv.org/abs/1905.04453

[18] Raoui, Y., Göller, M., Devy, M., Kerscher, T., Zöllner, J.M., Dillmann, R., Coustou, A.: Rfid-based topological and metrical self-localization in a structured environment. 2009 International Conference on Advanced Robotics pp. 1–6 (2009)

[19] Raoui, Y., Weber, C., Wermter, S.: Neoslam: Neural object slam for loop closure and navigation. In: Artificial Neural Networks and Machine Learning - ICANN 2022 - 31th International Conference on Artificial Neural Networks, Bristol, England, September 6-9, 2022, Proceedings, Part II (2022)

[20] Rolls, E., Stringer, S., Elliot, T.: Entorhinal cortex grid cells can map to hippocampal place cells by competitive learning. Network: Computation in Neural Systems 17, 447–465 (2006)

[21] Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: 2011 International Conference on Computer Vision. pp. 2564–2571 (2011). doi: 10.1109/ICCV.2011.6126544

[22] Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., Rus, D.: Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping (2020)

[23] Shan, T., Englot, B.J., Duarte, F., Ratti, C., Rus, D.: Robust place recognition using an imaging lidar. CoRR abs/2103.02111 (2021), https://arxiv.org/abs/2103.02111

[24] Tourani, S., Desai, D., Parihar, U.S., Garg, S., Sarvadevabhatla, R.K., Krishna, K.M.: Early bird: Loop closures from opposing viewpoints for perceptually-aliased indoor environments. CoRR abs/2010.01421 (2020), https://arxiv.org/abs/2010.01421

[25] Volkov, M., Rosman, G., Feldman, D., III, J.W.F., Rus, D.: Coresets for visual summarization with applications to loop closure. In: IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015. pp. 3638–3645. IEEE (2015). doi: 10.1109/ICRA.2015.7139704.

[26] Xiao, L., Wang, J., Qiu, X., Rong, Z., Zou, X.: Dynamic-slam: Semantic monocular visual localization and mapping based on deep learning in dynamic environment. Robotics Auton. Syst. 117, 1–16 (2019)

[27] Yongbao, A., Ting, R., Xiao-qiang, Y., Jia-lin, H., Lei, F., Jianbin, L., Ming, L.: Visual slam in dynamic environments based on object detection. Defence Technology (2020)

[28] Yu, F., Shang, J., Hu, Y., Milford, M.: Neuroslam: a brain-inspired slam system for 3d environments. Biological Cybernetics 113(5-6), 515–545 (December 2019). doi: 10.1007/s00422-019-00806-9, https://eprints.qut.edu.au/198104/

[29] Zhou, X., Weber, C., Wermter, S.: Robot localization and orientation detection based on place cells and head-direction cells. In: Lintas, A., Rovetta, S., Verschure, P.F.M.J., Villa, A.E.P. (eds.) Artificial Neural Networks and Machine Learning - ICANN 2017 - 26th International Conference on Artificial Neural Networks, Alghero, Italy, September 11-14, 2017, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10613, pp. 137–145. Springer (2017). doi: 10.1007/978-3-319-68600-4_17.