Multi-View V-SLAM

Shubham Priyadarshan, M.N Venkata Ramana and YSK SaratChandra (Team Halen)

***Abstract*— The current vSLAM methods are not making use of the data for the maximum extent possible and require some additional computation for camera distortions, our method mentioned in the paper aims at reducing the camera distortions to the minimum.**

I. INTRODUCTION

Visual simultaneous localization and mapping (vSLAM), refers to the process of calculating the position and orientation of a camera, with respect to its surroundings, while simultaneously mapping the environment. The process uses only visual inputs from the camera. Applications for visual SLAM include augmented reality, robotics, and autonomous driving.

Our proposed method is aimed at overcoming one of the aspect of traditional SLAM i.e., minimizing the lens distortion which leads to better results. This is achieved by using two cameras one of which is fixed and the other one revolving. We shall discuss the working in the future sections.

II. TRADITIONAL V-SLAM AND ITS CHALLENGES

A. Introduction and Procedure

The online robot estimation position from measurements of self-mapped features is a class of problem, in the robotics community, known as simultaneous localization and mapping (SLAM) problem. This technique consists in increasingly building a consistent map of the environment and, at the same time, localizing the robot's position while it explores its world. SLAM is perhaps the most fundamental problem to solve in robotics in order to build truly autonomous mobile robots.

Robot sensors have a large impact on the algorithm used in SLAM. Early SLAM approaches focused on the use of range sensors as sonar rings and lasers. Nevertheless there are some disadvantages associated with them.

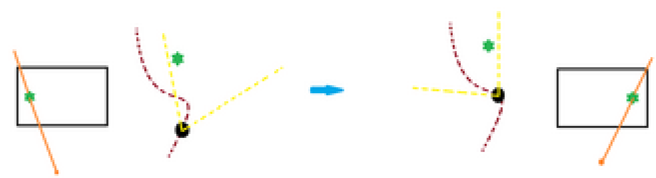
The aforementioned issues have propitiated that recent works move towards the use of cameras as the primary sensing mode. Cameras have become more and more interesting for the robotics research community, because they yield a lot of valuable information for data association.

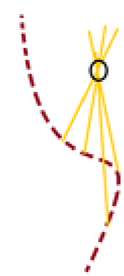
Simply put in words the robot collects visual data from camera and based on data and feature points extracted it predicts where is it, as it extracts more and more feature points,feature aligns the estimate keeps getting better and better. This data is again used to map out the surroundings.

B. Challenges

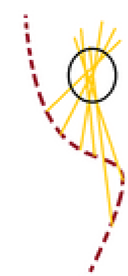
1. In traditional monocular V – SLAM, field of view is very limited. This cuts down the number of readings that can be taken when compared to a LiDAR sensor. Sometimes masking is done at the edges that reduces the FOV further.

2. We will need to do distortion correction on the camera, which makes it difficult to replace one camera with other, as most of our cameras are to be calibrated before–hand.





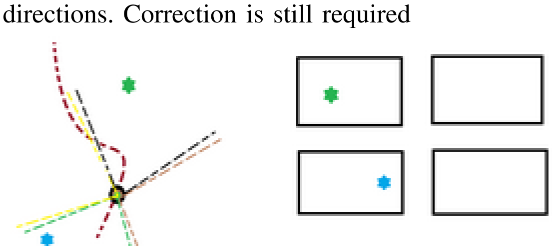
3. Ideal Result



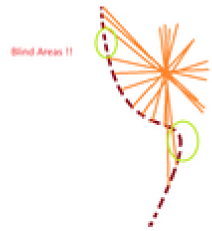
4. Practical Result

III. SOME SOLUTIONS

For increasing the accuracy: 1.Use cameras with higher FOV. 2. Use more than one fixed camera looking in different directions. Correction is still required



Results from approach through multiple fixed cameras.



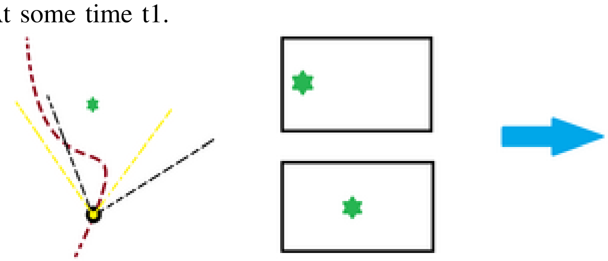
Some blind areas exist! Depending on the FOV of the camera and the number of cameras. FOV * n should be greater than or equal to 360 degrees for the best case scenario.

For Camera's distortion correction: 1.Get lens parameters from the camera manufacturer. Easiest way but proper information might not be always available. 2. Camera calibration using checker board. When we have complete control over the imaging process, the best way to perform calibration is to capture several images of an object or pattern of known dimensions from different view points. 3. Actively, try to estimate the straight lines in the image and try to find out the camera parameters. 4. When we have very little control over the imaging setup (e.g. we have a single image of the scene), it may still be possible to obtain calibration information of the camera using a Deep Learning based method.

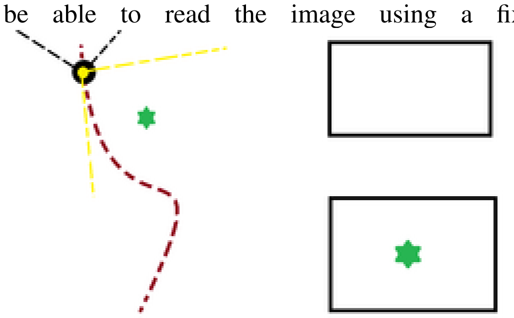
A. *Our Method*

We have tried to devise a method that can not only take more number of readings, therefore increasing accuracy, but also could eliminate the need to calibrate individual cameras, so that it can be a very generalized and out-of-the-box implementation. This is done by recentering the object to the center of the lens by moving the camera accordingly for a significant period of time.

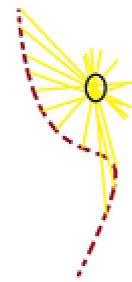
In the Figures, the black lines represent the FOV of a fixed camera whereas the yellow lines represent that of a moving camera.
At some time t1.



After some time, where the robot can no longer be able to read the image using a fixed camera.



Expected Result:



The accuracy of prediction is higher in this case.

Implementation is heavier than traditional monocular V-SLAM.

IV. Advantages

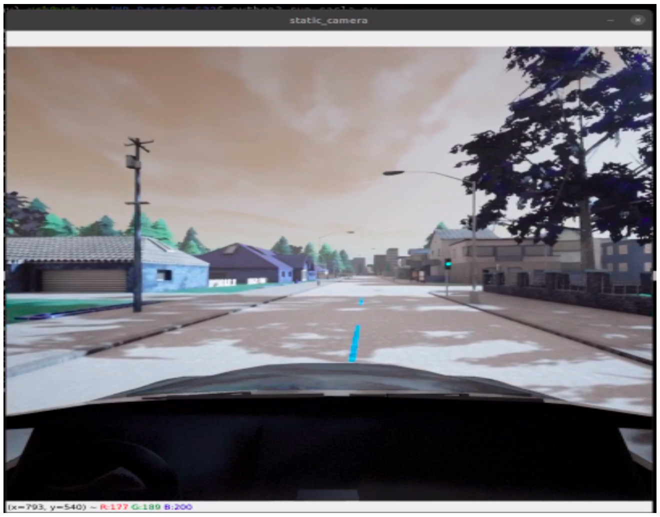
- FOV is not limited. Can capture more readings.
- FOV and distortion parameters of an individual camera don't matter in most cases.
- Therefore, can interchange cameras as per convenience.
- No blind areas.
- Can stack up more number of cameras on top of one another for maximum optimization.

  Error increases with increase in speed. But, on an average, accuracy is better than traditional V-SLAM.

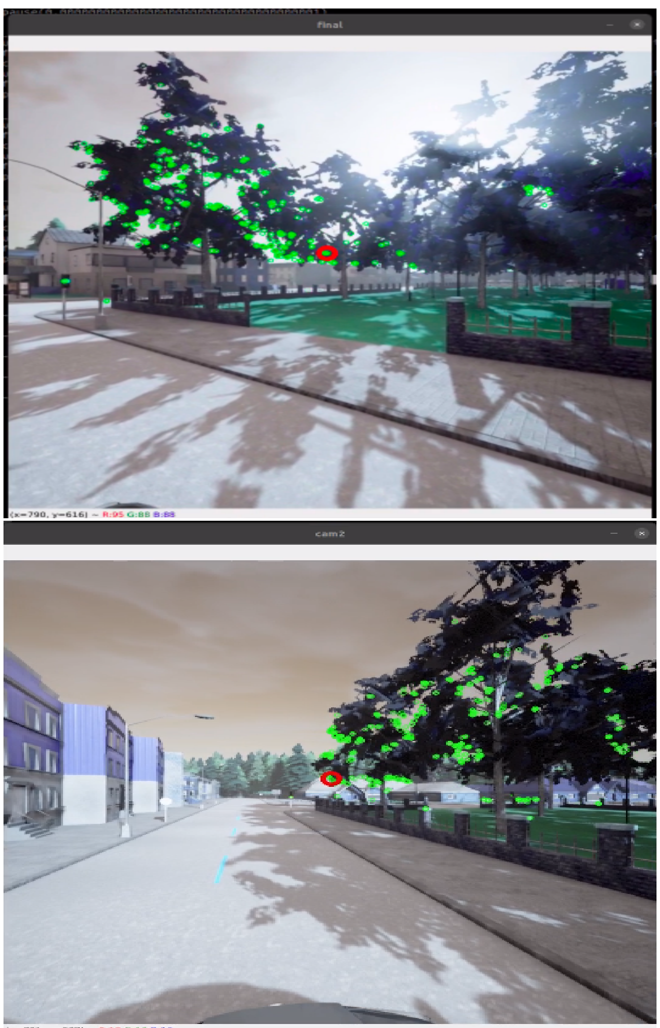
V. Implementation Methodology

- The experiment is simulated in Carla using a normal car with the cameras mounted.
- Carla simulator is used to simulate a car moving in an unknown world with a rotating camera at the top.
- The simulator was integrated with RoS to publish odometry of the vehicle.
- Feature extractor is used to identify landmarks and keep focussing on them such that the object is at camera's center.
- Tracked landmark positions are used to map out the environment.
- Using the tracked landmark positions we map out the environment.
- Can also be used as an SLAM input (ekf or any relevant variant).
- These points are repeatedly corrected by taking a mean of all the readings on them throughout the span of their focusing time.

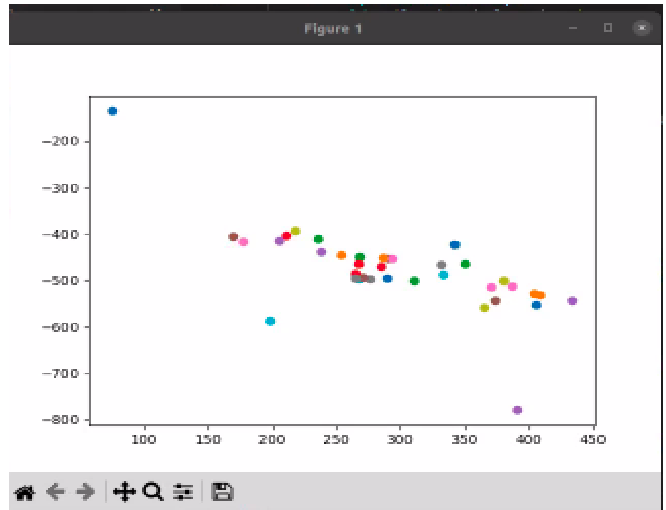
VI. Step Wise Simulation Results



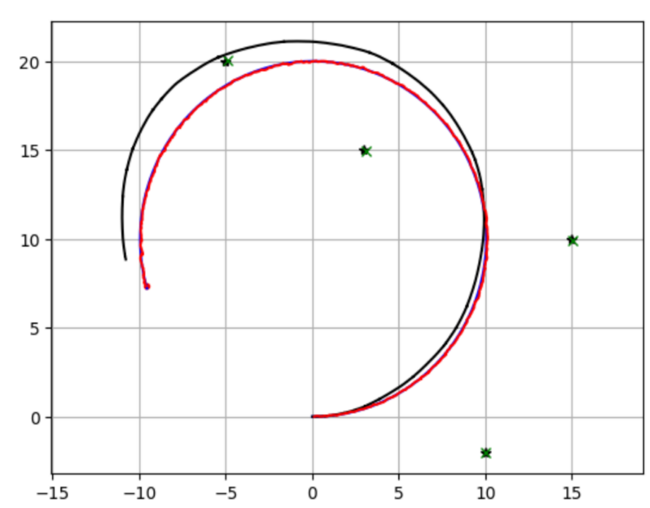
Static Camera view. Future Scope of the project includes using it to detect the next feature points to track and also give a general vSLAM output that can be used for general mapping . For now it is just to give the frontal view to the user.



Rotating Cameras. Used to select independant random feature points and keep tracking it while keeping them at the cameras' centres so that distortion is minimum. We implemented a PID controller to stabilize the movement of the cameras.



Final mapping of environment. It keeps getting updated as the car moves forward, the feature point location plotted finally is the average of all the positions recorded over time and once it is finally out of view or insignificant for the lens to detect its position is fixed and not updated further. After this a new feature point is focussed on.



Final EKF SLAM output for a constant velocity and angular velocity. The X marks are the position of the Landmarks and the blue line is the ground truth, red is the estimated path while the black plot is the dead reckoning path.

VII. Challenges Faced

- Using Gazebo to implement above scenario. To efficiently use gazebo we needed a high;y customized urdf file which was not available and couldn't be implemented due to time constraints. Hence the switch to CARLA was made.
- One more Challenge was to have a customized feature extractor which also outputs commands for the PID

controller.

- Final challenge was to implement SLAM. We chose the EKF slam, but one of the major challenges was to take care of rapid change in angular velocity around a sharp turn taken by the CAR. For the results we used constant velocity and a constant angular velocity.

VIII. Future Scopes

The implementation has manifold applications in the field of autonomous navigation and tracking. The principle with which it works, that is keeping the focus of camera fixed in order to get more number of readings. This improves accuracy overtime. The improved accuracy for the estimation of some points enabled us to use them as a reference points for unguided navigation and mapping of the vehicles/robots.

At last when combined with SLAM it can improve our estimates and thus improving our whole pipeline for SLAM.

We can stack up multiple rotating cameras to track more feature points simultaneously and further imporovise on localisation and mapping.
Though computationally more heavy it can improve on the accuracy.

IX. Conclusion

The traditional VSLAM has its own challenges with accuracy and needs some correction to overcome them. Our implementation on the other hand doesn't need any third party corrections to give a good result.
Our implementation only aims at improving accuracy and doesn't focus on the computation time efficiency, and hence has a lot of scope of improvement in the future. If kept on improvising it can be a very good alternative to the traditional VSLAM pipeline.

X. Acknowledgement

We would like to thank the Teaching Assistants of the course- Mobile Robotics and prof. Madhava Krishna for guiding us and providing us with necessary support at each step of the implementation.