ORB-SLAM using Multiple Cameras

Gautam Rajendrakumar Gare (*ggare*) , Shubhankar Potdar (*smpotdar*) Nithin Kumar Sathish(*nsatish*)
Shreyas Srivatchan (*ssrivatc*)

***Abstract*— Most of the autonomous systems such as self-driving vehicles, autonomous aerial systems have multiple cameras but typically use a single camera for SLAM. In this paper, we propose a different approach, where we present a Visual Simultaneous Localization and Mapping (VSLAM) system which integrates measurements from multiple cameras to achieve robust pose tracking and mapping for these autonomous systems in unknown complex environments. The well-known feature based ORB SLAM system has been extended to work with three cameras in the final implementation. The theory behind this system can easily be extended to multiple camera configurations. We have implemented this and shown experimental results on custom, KITTI, and the Argoverse datasets.**

I. INTRODUCTION

The current SLAM systems such as autonomous driving systems usually employ a multi-sensor/multi-camera fusion framework to overcome occlusions and other innate sensor shortcomings, like LIDAR sensors are sparse but very accurate and immune to lighting conditions as compared to visual camera sensors (RGB-D or stereo) which are dense but not accurate and susceptible to illumination variations. The current state-of-art feature-based SLAM algorithms such as ORB-SLAM [5] are designed to work with single camera system. So, there is a need for feature-based SLAM algorithms that can work with a multi-sensor/multi-camera framework. We try to address this need with our work.

A. Novelty about the approach

There is little literature where multiple cameras are used for ORB-SLAM. However, we found one literature where ORB-SLAM for fisheye camera has been proposed [7]. For our project, we adopt most of the methods and concepts of this paper.

The multi-keyframe proposed in [7] is a collection of multiple camera images which are used to generate feature descriptors that are used by SLAM tasks such as Loop closures. While calculating the feature descriptors the relative geometric configuration of the fixed multi-camera system (MCS) isn't taken into account. We plan on extending the multi-keyframe (MKF) with our novel unified-keyframe (UKF) approach which generates feature descriptors from the fused keyframe of all the cameras in the robot. A requirement for this is that the cameras have to possesses a shared field of view. We hypothesize that UKF would be a sparse representation that would provide performance boost while retaining the same level of accuracy as compared to MKF. We evaluate this hypothesis with our current work.

Some of the methods and concepts that we have used in our project are:

- Using multiple cameras for implementing ORB-SLAM.
- Using UKFs built using MKFs for Bundle adjustment.
- MultiCol - a generic method for bundle adjustment for multi-camera systems.
- Multi-camera loop closing.
- Evaluate different camera configurations for better performance.

B. Impacts of Proposed Research

The proposed research aims to explore ORB-SLAM [5] with multiple cameras. A direct impact of such an approach will be increasing the collective field-of-view (FOV) of the system. Additionally, there have been successful implementations where dense visual SLAM has been performed using multiple cameras. [4]. Such approaches tend to provide a more accurate estimation of the camera's pose when compared to the single camera approach. Therefore, through our project, by incorporating multiple cameras, we aim to improve mapping accuracy as well.

II. LITERATURE REVIEW

The ORB-SLAM algorithm proposed by Raul et. al. [5] was a landmark paper in feature-based SLAM techniques which achieved unprecedented accuracy with real-time CPU performance and lifelong operability and remains state-of-the-art algorithm to this date. ORB-SLAM calculates ORB features which are used for all the SLAM tasks (tracking, mapping, relocalization, and loop closing) which are operated in parallel, a method inspired from PTAM [3]. The algorithm uses a robust key-frame retention scheme which is generous in key-frame addition but restrictive in culling key-frames, this makes the SLAM system robust to fast camera motions and also enables lifelong operation. The ORB-SLAM algorithm was proposed for single monocular RGB camera system, which was further improved upon by the authors in ORB-SLAM2 [6] to work with a monocular, stereo or RGB-D cameras and with other additions such as map reuse, loop closing, and relocalization capabilities.

S. Urban and S.Hinz extend the ORB-SLAM to work with multi-fisheye camera systems in their work MULTICOL-SLAM [7]. They modified ORB-SLAM to work with Multi-keyframe which is a key-frame consisting of images from multiple fisheye cameras and performed pose estimation using MultiCol [9] which is a modular Bundle Adjustment (BA) technique for multi-camera systems, a prior work of the authors. They also introduce multi-camera loop closing, a minimal non-central absolute pose estimation and different initialization techniques in their work. Our work builds on top of this, by extending the multi-keyframe with our novel unified-keyframe and coupled with other optimizations and improvements.

III. Theory Developed

The main goal behind the project was to explore ways to implement SLAM using multiple cameras. In order to achieve this, methods from [7] were adopted. Specifically, a multi-keyframe (MKF) concept which uses as many frames as cameras was used for efficient tracking. Corresponding modifications to Bundle adjustments (MultiCol) and multi camera loop closing were also used. Though this method seemed to solve the purpose, it possessed some limitations such as rotational drift and an algorithm which decreased in performance as the number of cameras increased. In order to overcome this, we developed another approach which fuses the images of all the cameras using image stitching using homograph matrices. This would generate a single image that can in turn be used on naive ORB SLAM. We hypothesised that using a fused image would provide features that were more sparse than MKF leading to a better SLAM functioning overall. Subsequent sections show results of the MKF and UKF implementation.

IV. Dataset

For the single camera ORB-SLAM system we are testing it on two datasets: Custom dataset and KITTI dataset.

We collected our own data using an Intel D435 realsense depth camera. The camera was placed on a setup and moved around the house (living room and dining area) in order to maintain a constant height. The results section shows the created 2d pointcloud resulting from ORB SLAM performed on that data.

The KITTI dataset[2] contains stereo sequences recorded from a car in urban and highway environments. The stereo sensor has 54 cm baseline and works at 10Hz with a resolution after rectification of 1240 X 376 pixels.

For the multi-camera ORB-SLAM system we use Argoverse dataset [1]. The full Argoverse dataset was collected using 2 LiDAR's, 7 ring cameras and 2 forward stereo cameras. In this project, we are only using the center, left and right ring cameras and ignoring other ring camera and the LiDAR data. The stereo cameras works at 5Hz with a resolution of 2056 X 2464.

V. Code Implemented

In this project, we have three algorithms. The first one is the ORB-SLAM2 algorithm by [5]. In order to implement this, we used the GitHub implementation of the algorithm[1] and used the KITTI dataset and the Argoverse dataset as input. In order to implement the MKF algorithm, we used the code derived from the corresponding GitHub implementation [2]. For this implementation, we used our custom dataset that was collected as well as the Lafida - A Laser scanner Multi-Fisheye Camera Dataset which includes frames from three fisheye cameras [8]. For the UKF implementation, we implemented our own code for the image fusion portion. The corresponding git link can be found in the footnote[3]. We have used the Argoverse dataset for this implementation as well.

VI. Results

A. ORB SLAM2

In order to have a base implementation, we implemented the ORB SLAM2 on the KITTI and Argoverse dataset. The corresponding trajectory that was tracked and a screenshot of the ORB features are shown in figures 1 and 2. The tracking here uses images from one source camera as the input. A sample video of ORB SLAM2 working on the KITTI dataset can be viewed from the footnote below [4]. Additionally, a sample video of ORB SLAM2 working on the Argoverse dataset can also be viewed from the footnote below [5]. (Please copy paste the link. Clicking on the link will not work as the second line is not copied to the new tab created).

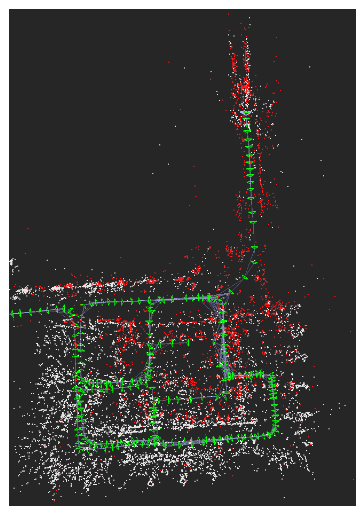

Fig. 1. Generated trajectory and map using ORB-SLAM

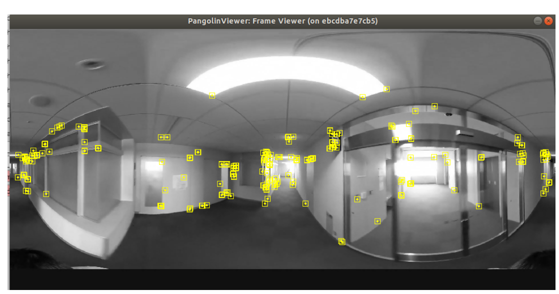

Fig. 2. Generated trajectory and map using ORB-SLAM

[1] https://github.com/raulmur/ORB_SLAM2
[2] https://github.com/urbste/MultiCol-SLAM
[3] https://github.com/ggare-cmu/16833_Project_UKF_SLAM
[4] https://drive.google.com/file/d/1O8meVzEbH1f7MlZzOhv4GNnGSqAOpii2/view?usp=sharing
[5] https://drive.google.com/file/d/1jxSTmJeOmNzaprPk3GPdXSc3aGVb9amz/view?usp=sharing

B. MultiCol SLam

The MultiCol SLAM is an implementation of [9] which uses the concept of Multi Keyframes. The dataset used was [8] which includes views from three fisheye cameras. A screenshot of the keyframes generated and features tracked are shown in figures 3, 4 and 5. Figure 4 specifically shows the placement of the three cameras (fisheye) and their relative positions with respect to each other and the body frame, of the setup on which they are placed. A sample video of the MKF working on the multi-fisheye dataset can be viewed from the link in the footnote [6]. (Please copy paste the link. Clicking on the link will not work as the second line is not copied to the new tab created).

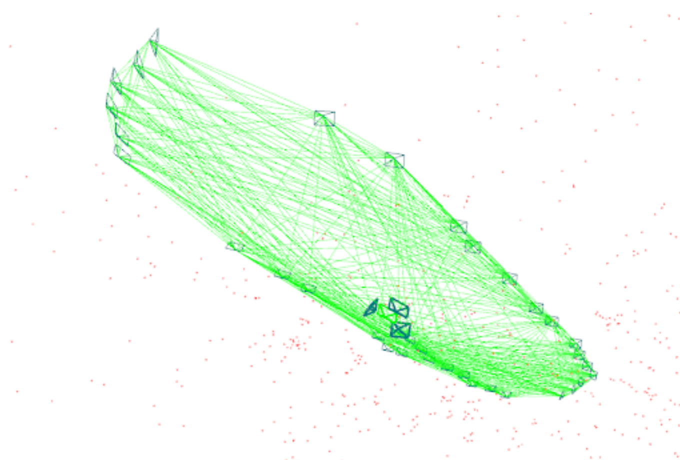

Fig. 3. Co-visibility graph of keyframes

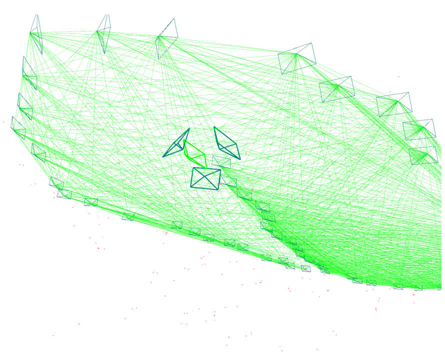

Fig. 4. Zoomed co-visibilty graph showing camera configuration

C. Unified Key Frame SLAM

As an alternative to the MKF approach, we implemented UKF where the frames from multiple cameras were fused. Once they were fused, we used ORB-SLAM on that to perform localization. This is an alternative way to implement ORB SLAM where advantages of using multiple cameras such as increased field of view are derived whereas it is not essential to track Multiple keyframes and perform MultiCol bundle adjustment, thereby reducing the overall complexity of the SLAM system. The figures 6 and 7 show the separate camera views and the merged views respectively. Additionally, the figures 9 and 8 show the features tracked and the resulting trajectory from UFK on the Argoverse dataset. Here, we used the Argoverse dataset so that we can use inputs from multiple cameras. A sample video of the UKF working on the Argoverse dataset can be viewed from the link in the footnote [7]. (Please copy paste the link. Clicking on the link will not work as the second line is not copied to the new tab created).

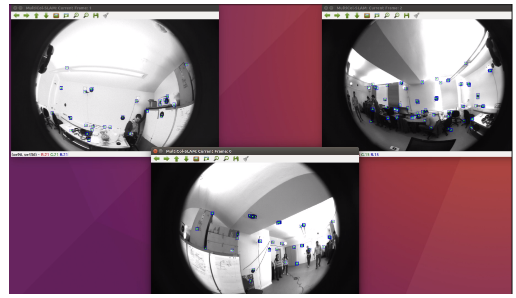

Fig. 5. Image showing features simultaneously tracked on three fisheye cameras

(a) (b) (c)

Fig. 6. images of the left, center and right views of three cameras

Fig. 7. merged view of all the three cameras

VII. Discussion of Results

We obtained satisfactory results for the UKF, MultiCol (MKF) and ORB SLAM systems. We also observed that each method has some advantages that others did not.

A. MultiCol SLAM (MKF) vs ORB SLAM

There are certain obvious advantages to MultiCol SLAM over ORB SLAM. The first one being an increased field of view. In the implementation, three fisheye cameras were used which resulted in almost 270 degrees of collective field of view. However, increasing the number of cameras also resulted in a decrease in speed. Extending this method to

[6]https://drive.google.com/file/d/1MWhRgG8FV7jGG5r_-qBQJeiwrhKt65mn/view?usp=sharing

[7]https://drive.google.com/file/d/19pL-VZXd_0G6H3jgwbmglU0buxwD2b4J/view?usp=sharing

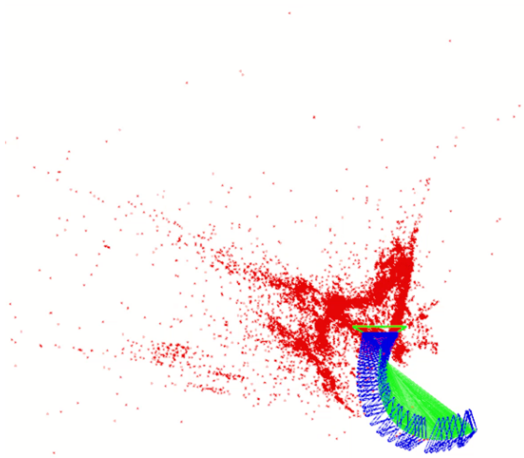

Fig. 8. Generated trajectory and map using UKF

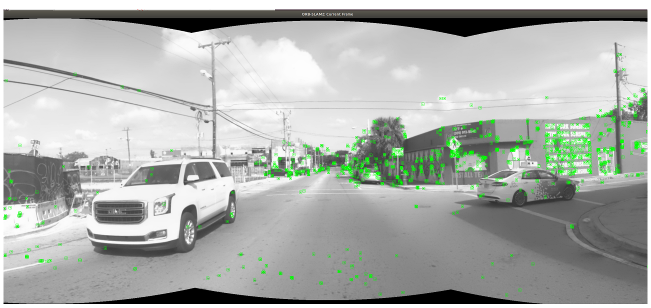

Fig. 9. Features tracked using UKF

work with many cameras (greater than 5 or 6) might prove intractable for a computer with low computing capabilities. Another factor that impacts the performance is the camera placement as the MultiCol MKF system is prone to rotational errors. An efficient camera placement might alleviate this issue.

B. ORB SLAM vs UKF

We did a comparison between the UKF approach and conventional ORB-SLAM on the Argoverse dataset. By doing so, we found that ORB SLAM performed better in areas where the trajectory was straight whereas the UKF performed better in scenarios where sharp turns existed. We believe this observation is because of the wide view which the system has while taking turns which allows it to extract more features and form a better map. With respect to performance, we found that conventional ORB-SLAM performed better when the trajectory was straight. This observation was contrary to our hypothesis. One possible reason is that the stitching of images across multiple time steps is not consistent i.e. certain sections of the image across frames deviate from their position. Though there were no significant deviations from the ground truth, we noticed that the map from ORB SLAM2 was better aligned with the ground truth.

C. UKF vs MKF

Both the UKF and MKF were similar in performance w.r.t localisation. However, the UKF method was slightly faster than the MKF as UKF essentially has one stitched image as input whereas the MKF has as many inputs as cameras. We also noticed that UKF functions better during sharp turns possibly because of the increased field of view that essentially results in a large number of features. The main disadvantage of UKF over MKF is that UKF has a requirement that cameras that are used must share a FOV. However, there is no such requirement for MKF. This might prove to be a disadvantage as there might be robotic platforms that will not be able to afford multiple cameras that also share a field of view.

D. Summary

Each of the methods possesses one or more advantages over the other. While UKF performs well in sharper turns, naive ORB SLAM performs better in scenarios where the trajectory is straight. The MKF can be used in scenarios where there are limited number of cameras (up till 4) and a computer with good computing resources. However, this situation is especially useful in scenarios where limited number of cameras can be used to obtain a collective FOV more than about 150 degrees. In areas or maps where there are a lot of twists and turns, MKF SLAM might provide an inaccurate map as the method is prone to rotational drifts thereby leading to localization inaccuracies and corresponding errors.

VIII. Challenges Faced

One of the main challenges we faced was to get a dataset which had 3 or more camera views that can be used for our SLAM implementation. Our initial plan was to collect a custom dataset on campus that would be big enough to obtain significant results, but we had to alter our plans due to obvious reasons. We have managed to collect our own dataset, but it was collected at an apartment due to which the size of the dataset is very small. As alternatives, we have relied on the Lafida [8] dataset which provides images from three fisheye cameras. We were able to use this for the MultiCol implementation. Additionally for the UKF implementation, we needed multiple images that were not fisheye views. As a result, we used that Argoverse [1] dataset that provides a small dataset containing views from multiple perspective cameras. We were able to use this effectively for the UKF implementation. Other challenges involved understanding the theory of using MKF as it was novel. Understanding corresponding modifications to bundle adjustment and loop closing was challenging as well.

A. Performance with respect to timeline

During the initial stages of the project, we could keep up with the timeline we proposed with very small delays due to schedules of all our group members. However, a significant shift in our timetable occurred due to the campus lockdown. This resulted in us having to alter our data collection pipeline as we had initially planned to collect our own dataset. Subsequently we decided to use the [8] and [1] dataset for our project. Due to these delays, we had to work more than we planned in the month of April. However, subsequently, we achieved all the goals we had initially set for our project.

IX. Conclusion

In this project, we have explored ways to implement SLAM using multiple cameras. Specifically, we have explored MultiCol SLAM and implemented our novel UKF method which fuses images to generate a keyframe and uses those for localization and mapping. Through our experiments, we find that UKF performs better than the other methods in areas of turns but has a decreased performance in straight paths. We observed MultiCol SLAM (MKF) performed satisfactorily but had some limitations as well. An area of study we feel will be beneficial to both UKF and MKF implementations will be a thorough mathematical analysis of camera placements and corresponding factors such as the amount of shared Field of View etc. We believe that such an analysis will prove to be extremely beneficial for multiple camera SLAM systems. Also more work on a better image stitching algorithm that takes into account camera positions will lead to better performance of the overall SLAM system.

References

[1] Ming-Fang Chang, John W Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[2] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[3] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.

[4] Xinrui Meng, Wei Gao, and Zhanyi hu. Dense rgb-d slam with multiple cameras. *Sensors*, 18:2118, 07 2018.

[5] Montiel J. M. M. Mur-Artal, Raúl and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[6] Raul Mur-Artal and Juan D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33:1255–1262, 2017.

[7] Steffen Urban and Stefan Hinz. Multicol-slam - a modular real-time multi-camera slam system. 10 2016.

[8] Steffen Urban and Boris Jutzi. Lafida—a laserscanner multi-fisheye camera dataset. *Journal of Imaging*, 3:5, 01 2017.

[9] Steffen Urban, Sven Wursthorn, Jens Leitloff, and Stefan Hinz. Multicol bundle adjustment: A generic method for pose estimation, simultaneous self-calibration and reconstruction for arbitrary multi-camera systems. *Int. J. Comput. Vision*, 121(2):234–252, January 2017.