VISUAL ODOMETRY BY INTEL REAL SENSE T265 BY 3 DIFFERENT VIEWS

[1]SHIVANI BALDWA, [2]RAGHAV JETHLIYA, [3]CHUANG JAN CHANG

Ming Chi University of Technology, Taiwan,
No. 84, Gongzhuan Road, Taishan District, New Taipei City, 243
E - Mail: [1]baldwashivani@gmail.com, [2]raghavmaheshwari.905@rediffmail.com, [3]cjchng@mail.mcut.edu.tw

Abstract - MOIL-SDK can reference different spatial angles to distill multiple-images conforming to those of central perspective from a fisheye image. Accordingly, we extend an open monocular visual odometry project to generate 3 odometry that tracks from a fisheye video. It is a real-time, precise, and persistent approach. The contributions of our work are to create a MOIL video dataset using Intel RealSense T265 camera as it associated with an embedded IMU sensor that produces 6DOF that helps to generate ground truth data for the camera moving. We also modify the regular single-tracked video odometer to multiple tracks. With this extension, it is forecast able that the inherent error accumulation issue consults to a single trajectory will have a chance to improve.

Keywords - MOIL-SDK, Intel RealSense T265, Visual Odometry: Fisheye image, Camera Calibration.

I. INTRODUCTION

Nowadays, Visual Odometry is been emerging motive in the field of computer vision. Visual Odometry which means estimating the movement of the camera in real-time using series of images i.e. also referred to as ego-motion. Here, odometry is invented by a Greek word which has a meaning “Route Measure" [4]. There are various applications in the field of autonomous driving, robotics, wearable computing, augmented reality, drone controls, and many others. After we have camera mounted on objects which might be car, robot, or the other thing we want to create a 6 DOF (Degree of Freedom) trajectory with assistance of video stream or sequence of images which we are getting from the camera. Ultimately, we need to verify the position and orientation of a camera by which camera images are analyzed. Odometry can be done by both stereo and monocular cameras. When we use a single camera for motion estimation is defined as Monocular Camera, whereas using two or more cameras for motion estimation referred to as Stereo Camera. Comparison to stereo VO, monocular VO is becoming more trending and interesting among researchers, as monocular cameras are cheaper and easy to calibrate and suitable for both indoors and outdoors environments. But there are many problems associated with monocular VO the one is accuracy, and scale estimation. In our approach we are using monocular VO with Intel T265 camera.

1.1 Dataset

There are many datasets available to perform odometry like: KITTI, EUROC, TUM RGB-D etc. This dataset contains the set of calibrated images with ground truth data. But creating our own dataset is a challenging task. We have compared different methods used in odometry with our created methodology. KITTI dataset is most often used dataset as it has set of pinhole images, ground truth data, calibration file and timestamp to perform odometry. Ground truth data defines the rotation and translation for data defines the rotation and the translation for camera which is recorded during creation of dataset and after that it is used to evaluate the estimated path odometry.

In our approach we created our own dataset named MOIL dataset, with the help of Intel T265 camera which has an inbuilt IMU sensor which helps to create the 6 DOF and extract the ground truth data. We captured a set of fisheye images with their respective ground truth data to perform odometry.

1.2 Feature Detection & Feature Matching

As Feature matching is the important factor to perform odometry. There are various algorithms to accomplish feature matching and odometry. It is done between pair of correspondence images as shown in below Figure 1.

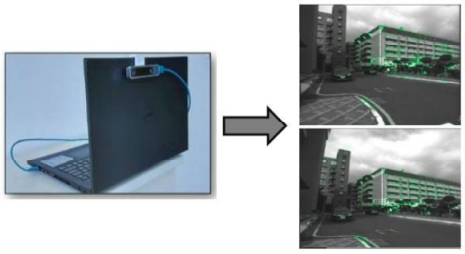

Figure.1 Input from T265 camera and extracts features between images.

In our approach we used FAST feature detector algorithm. As there are many feature matching algorithms available like: SIFT, SURF, ORB etc. but for real time application they are unable to give better performance. FAST is introduced as corner detector algorithm, which could also use for feature point extraction. FAST has more computational efficiency than any other method and if machine learning

techniques are used with this than it gives excellent performance in terms of computational efficiency and resources. It is high performance algorithm most suitable for real time application in computer vision.

II. RELATED WORK:

Investigation on visual odometry is exercised from way back. The advantage of using a wide-angle or fisheye lens is the larger field of view. While doing this we suffer from distortion which makes straight lines appear curved line also impacts the image. The goal of calibration is to accurately measure the intrinsic and extrinsic parameters of the camera and remove the distortion from the images.

2.1 Calibration

The intrinsic parameters refer to estimate the set of parameters that are internal and fixed in a camera model. It includes focal length, principal point, and optical center also skew coefficient. It also allows mapping between camera coordinates and pixel coordinates. The extrinsic parameters refer to camera parameters that are external to the camera and may change for world coordinates. It consists of rotation, R and translation vector, t. It defines the location and orientation of the camera in the world coordinate system. To compute calibration result accurately the user should take many checkerboard images and should know the size of each square. Later, should perform camera calibration by any toolbox such as MATLAB, OpenCV [3] Motion estimation is core computational step performed in every image, for that existing approaches can be classified feature-based approach and direct based approach. The extrinsic parameters refer to camera parameters that are external to the camera and may change for world coordinates. It consists of rotation, R, and translation vector, t. It defines the location and orientation of the camera in the world coordinate system. To compute calibration result accurately the user should take many checkerboard images and should know the size of each square. Later, should perform camera calibration by any toolbox such as MATLAB, OpenCV [3] [7].

2.2 Overview of VO System

Motion estimation is core computational step performed in every image, for those existing approaches can be classified feature-based approach and direct based approach. Feature-Based approaches are majorly used for object recognition, Motion tracking, 3D reconstruction, image alignment and robot navigation. The feature is a piece of information which is relevant to solve computational task it can be a specific structure in an image such as points, edges, corners, or objects. Features are always matched between the pair of images based on their orientation and local appearance. [5] In the feature-based approach we need to extract the feature points between the first image and second image in such a way that their corresponding points should be matched. However, not all feature points are favorable we need to distinguish between the matched points and unmatched points. [1] We can ignore the unmatchable points as they affect performance efficiency. Figure 2 shows us overview illustrated for VO system; it has four steps such as feature matching, feature detection, feature tracking, and motion estimation. [2] Feature matching is a method of detecting a set of interest points in an image, once the features are extracted from two or more images then we establish some preliminary feature matched between these images.[8] We match all features that are within the disparity limit from each other. Most commonly used computer vision algorithm is Harris Corner Detection which extracts the corner and deduces feature of an image. Feature-Based approaches are majorly used for image alignment, object recognition, Motion tracking, and robot navigation. The feature is a piece of information which is relevant to solve computational task it can be a specific structure in an image such as points, edges, corners, or objects. Features are always matched between the pair of images based on their orientation and local appearance. [5]

2.3 Intel RealSense T265

Intel Technology has wide range of cameras which include depth and tracking technology, here we are using the RealSense T265 Tracking camera. It consists of two wide angle lenses, an Inertial Measurement Unit IMU sensor and two Visual Processing Unit VPU. The field of view can be measured horizontally, vertically or diagonally. It produces 6-DOF, Degree of Freedom also provides the sample rate of 200Hz with 30fps.It is preferable to work well in lit environment and can sustain well in lower light levels. It is small in size and light weight camera which has high precision tracking. T265 is supported by 'librealsense' on Windows and Linux operating. This is commonly used in robots, drones, augmented reality and virtual reality.

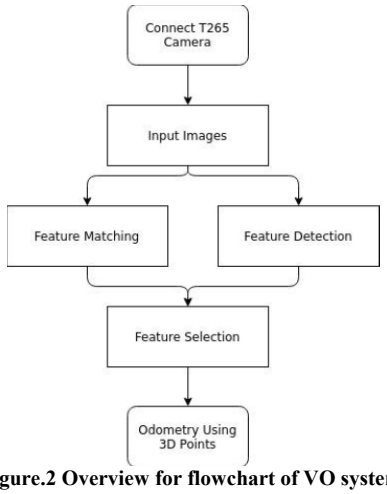

Figure.2 Overview for flowchart of VO system

III. THE PROPOSED APPROACH

In this paper, our proposed approach shows 3 view odometry with different view angles by using MOIL SDK. MOIL SDK is a collection of functions support C++ developments for fisheye image applications. This library can extract multiple views with different angles; here alpha and beta (α, β) are parameters responsible for the extraction of view with any angle. In this approach we used an OpenCV calibration model for calibration to get extrinsic and intrinsic parameters which are later used for further odometry process.

Traditional approach generates one view, whereas in our method we generate three views the purpose of for 3-views is to generate a pair of X-Y Maps for the specified alpha, beta and zoom parameters, then the result X-Y Maps can be used later to remap the original fisheye image to the target angle image. The result rotation is beta Offset degree rotation around the Z-axis(roll) after alpha Offset degree rotation around the X-axis(pitch). Traditional approach for Odometry produces only single view that is front view by source pinhole images, whereas in comparison to our approach we extract 3 view angles from Source fisheye images as fisheye image provides a wide view angle so we can extract multiple view from it. As MOIL SDK provides functionality to extract multiple views with any angle. Before, Monocular Odometry uses in Robots, in autonomous driving vehicles, navigation etc. But can only have Front view and its trajectory. This paper proposed 3-views with Fisheye images which will increase the efficiency of monocular odometry to next level, using this method in Odometry application will provide users to survey multiple angles and multiple trajectories for any specified angle simultaneously.

IV. RESULT & DISCUSSION

In this section we have proposed our visual odometry system. Here all experiments are performed on Desktop with Intel Core i5-9400 CPU @ 2.90GHz processor and 16GB RAM running on Ubuntu Linux 18.04.4 LTS.

We have evaluated the proposed approach with different sequences, which are tested in outdoor environments. We perform experiment on MOIL dataset containing 1844 frames. The system is applied with the assist of C++ and OpenCV. We have analyzed our VO system on proposed MOIL dataset. We are introducing a new method in which users can perform odometry using 3 views i.e. Front, Left, and Right view simultaneously. It also can run a trajectory for all 3-views using source fisheye images. To generate a 3-view trajectory we have integrated our Visual Odometry solution with the MOIL-SDK which is developed by MCUT Omni directional Imaging Lab, Taiwan. After integrating odometry with MOIL SDK, our solution can automatically undistort the fisheye images i.e. convert images to a pinhole model and can generate 3 different views with different angles(angles can be modified as required) and generate 3 view odometry for all respective views(Front, Left, Right) simultaneously. Figure.4 and Figure.5 represents 3-view trajectory produced with the help of ground truth data and sequence of images. Comparing it with traditional approach in which we only get one view, we cannot differentiate it on basis of different view angle, zoom factor and feature points.

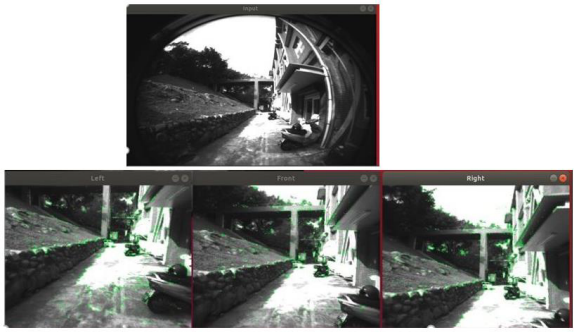

Figure.3 Images shows when the trajectory is calculated, it changes original fisheye images to Front, Left and Right view.

After examining the result, Figure 4 and 5 our approach shows better and promising result in the area where is more texture, even surface, even visibility and sunlight which helps to detect more feature points. It is also dependent on Zoom factor as increasing the Zoom angle will affect the matching points between images.

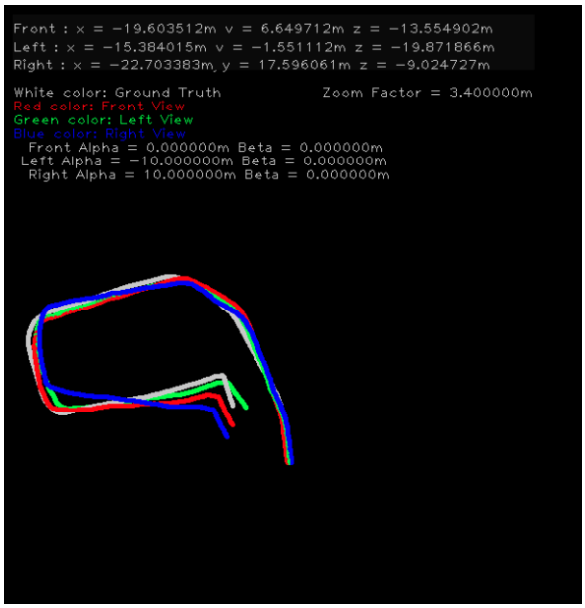

Figure.4 Three different views trajectories from source fisheye image, where (α, β) varies for Front, Left and Right view as (0°,0°), (-10°,0°) and (10°,0°) respectively. Here zoom factor is increased to 3.8 and can see dispersion in resultant trajectory which is due to less feature matching between images.

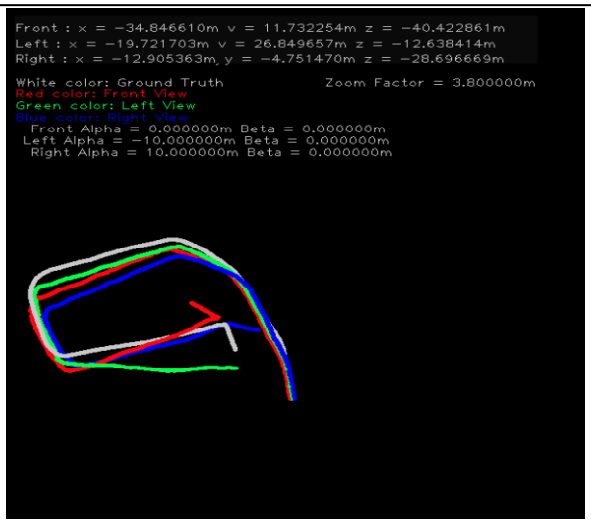

Figure.5 Three different views trajectories from source fisheye image, where (α, β) varies for Front, Left and Right view as (0°,0°), (-10°,0°) and (10°,0°) respectively. Here, zoom factor is changed to 3.4 and can examine the better resultant trajectory, due to more feature matching between images.

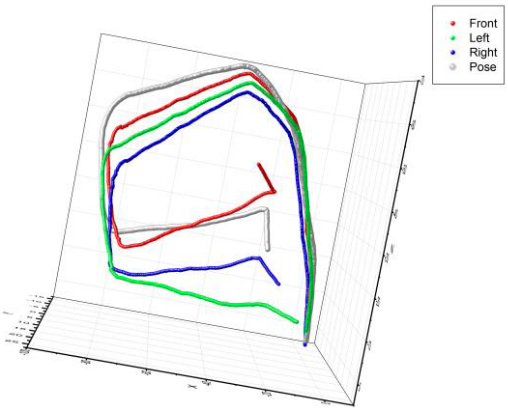

Figure.6 3D space graph For Trajectory shown in Figure 4

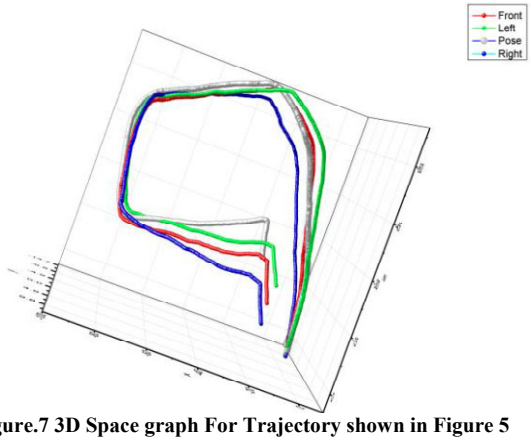

Figure.7 3D Space graph For Trajectory shown in Figure 5

V. CONCLUSION & FUTURE WORK

In this paper we have constructed odometry with 3 views using MOIL Library. Firstly, we did calibration for the camera to get the camera extrinsic and intrinsic parameters. Secondly, we created our dataset with the help of Intel T265 (Tracking camera) and inbuilt IMU sensor to get 6DOF (Degree of Freedom), and then last but not the least we implemented Monocular visual odometry with 3 views. Our plan to extend this project to get 6 views odometry and define a method to discard the unmatchable feature point from all the views which will increase the performance efficiency of our system and later we try to implement a User Interface to make our system more convenient and user friendly.

REFERENCES

[1] Wang Zhou, Hao Fu, Xiangjing An: "A Classification-Based Visual Odometry on Intelligent Human-Machine Systems and Cybernetics".

[2] DING Lianghong, WANG Runxiao, KI Tao: "Visual Odometry Research Based on Efficient Feature Tracking"

[3] Davide Scaramuzza , Friedrich Fraundorfer: "Visual Odometry Part 1, The First 30 Years and Fundamental", IEEE Robotics & Automation Engine.

[4] Mohammad O.A Aquel, Mohammad H. Marhaban, M. Iqbal Saripan, Napsiah Bt. Ismail: "Review of Visual Odometry: types, approaches, challenges, and applications."

[5] David Nister, Oleg Naroditsky, James Bergen: "Visual Odometry" , Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition

[6] Minjie Lin, Qixin Cao, Haoruo Zhang : "PVO: Panoramic Visual Odometry", 2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARAM)

[7] Andre Silva Aguiar, Filipe Nevas Dos Santos , Luis Santos, Armando Jorge Sousa: "Monocluar Visual Odometry Using Fisheye Lens Camera"..

[8] Shashi Poddar, Rahul Kottath, Vinod Karar: "Evolution of Visual Odometry Techniques"

★★★