Monocular Visual Odometry on Blimp Platform

ECE4012 Senior Design Project

Section L5B, XXLs Team
Project Advisor, Dr. Zhang, Fumin

Fanzhe Lyu, fanzhe@gatech.edu@gatech.edu, Team Leader
Ruoyang Xu, rxu76@gatech.edu
Yilun Xie, yxie306@gatech.edu
Yifan Shen, aelitashen@gatech.edu

Submitted

2020 May 3

Executive Summary

This project was motivated to extend capabilities of the current blimp (GT-MAB) platform on localization outside the lab environment, in real-life general scenarios using visual odometry techniques. Current blimp platform (GT-MAB) does not have capability to localize itself without an OptiTrack system. The installation of the OptiTrack system restricted the circumstances where it can operate. The dynamics of the blimp is unique and the carrying capabilities of the blimp limits the hardware that could be put on board. We designed a software stack for visual odometry that is ROS (Robot Operating System) compatible. The entire pipeline addresses issues on image noise, provide a rough estimation on the current pose of the blimp and provide a visualization of the trajectory that blimp has been traveled. The benefit of the system enables the blimp to complete more supplicated task such as autonomous navigation, and robot swarms. GT-MABs are light weight, safe and human-friendly, and with the visual odometry, more applications such as robot tour guide, warehouse surveillance system can be easily built upon. The costs of a blimp system are around $200, and the software system is required to run on computer with appropriate computation power.

Nomenclature:

Acronyms	**Definition**
DSO	**Direct Sparse Odometry**
FOV	**Field of View**
GHz	**Gigahertz / 10^9 Hertz**
GT-MAB	**Georgia Tech Miniature Aerial Blimp**
IMU	**Inertial Measurement Unit**
LK	**Lucas-Kanade Method**
MSE	**Main Squared Error**
ROS	**Robot Operating System**

SLAM	**Simultaneous Localization and Mapping**
VO	**Visual Odometry**

Table of Contents

Monocular Visual Odometry on Blimp Platform with Noisy Image

1. Introduction

The project is deploying and achieving robust visual odometry for robotic blimps to estimate the position and orientation through a monocular camera. The system consists of a robotic blimp, a monocular camera and a 5.8GHz transceiver system. The system cost around $100 in total, and with probable replacement costs for malfunction parts caused by control failures of the aerial robot, any team wish to reproduce the result with the system should requested $200 to develop the system.

1.1 Objective

This project is designed to be an upgrade to the existing Blimp platform. The objective is to deploy and achieve accurate visual odometry on Blimp, given the constraints of camera image quality and the platform dynamics characteristics. The visual odometry system will provide an accurate estimation of the robot pose, including position and orientation of the blimp. The system will enable more complicated robot actions such as autonomous navigations, and swarm robots [1]. Combining the safety feature of the blimp. Applications such as robot tour guide can be built upon, in the context today, it can be safely used to guide people at COVID-19 test sites.

1.2 Motivation

The current blimp platform (GT-MAB) does not have a localization capability without the OptiTrack system. The OptiTrack system is an absolute localization system [2] that requires prior setups and are hard to be used when the environment contains obstructions to its IR receivers. Such limited localization ability will only allow the blimps to operate in rooms with OptiTrack system installed, and restrain the blimp for more useful applications such as robot tour guides, warehouse surveillance, etc. It is crucial to extend blimp's capability to operate without an OptiTrack system.

However, the dynamics of the blimp platform is unique and ruled out open-source vanilla VO (Visual Odometry) implementation. In addition, carrying capability of the blimp limits the camera that provides high quality images. The blimp dynamics and low-quality analog camera require a custom-built software pipeline to enable localization with visual odometry on the blimp.

In summary, we would like to extend the blimp localization capability outside the lab given the blimp dynamics and hardware limitations. This project will enable roboticists as well as everyone who uses blimp to build more sophisticated applications such as autonomous navigations.

1.3 <u>Background</u>

VO (Visual Odometry), SLAM (Simultaneous Localization and Mapping) are classic topics studied in 3D computer vision and have been studied decades. However, no go-to solutions have been invented yet. Traditional and popular methods of performing VO, SLAM is suing LK (optical flow) to estimate and construct structure of motion. Current Methods include EKF-SLAM, ORB-SLAM, SVO, ROVIO, and DSO (Direct Sparse Odometry) [3, 4, 5, 6, 7]. However, due to technical debt of the methods, most of existing code cannot be used open box.

Common methods are usually divided into two classes: direct methods and feature based methods. Direct methods estimate how the structure from motion changes using variants of LK, minimizing photometric reprojection errors, while feature based methods apply feature extraction first then minimizing the transformation errors. Bundle adjustments and optimization are widely used in SLAM and VO projects.

However, most of the existing VO and SLAM system relies on high resolution, dedicated camera. Few systems have been tested working on low quality cameras such as camera on phones or analog cameras.

2. Project Description and Goals

The project enables the blimp platform to localize itself in real-time using a monocular camera and without the help from an OptiTrack system. The project includes an end to end solution from image processing to the odometry estimation. The project delivered a pipeline of software packages that performs robust visual odometry in real time on the blimp platform. We delivered a hardware-software system including:

- Noise reduction of camera stream and rejection of unrepairable image frames
- Fisheye Camera Calibration
- Visual odometry
- Visualization of estimated and ground truth trajectory in ROS
- Benchmarking Tools

The final product can estimate the path of the blimp in an arbitrary unknown environment using a single camera. We also provide visualization tool and benchmarking tool to help visualize and evaluate performance of the system. The target user of the project is roboticist and any other engineers who would like to build a system or applications that require to localize the robot without the help from an OptiTrack system.

3. Technical Specifications & Verification (HAVEN'T STARTED)

The two primary specifications for the project would be estimator computation time and maximum deviation distance. Table 1 contains the relevant specifications for those two aspects. Consider that blimps travel at a maximum speed of 0.5m/s and a desired maximum update distance to be 2.5cm. This would derive the odometry update frequency to be around 20Hz, which leads to a maximum computation time of 50ms. The maximum deviation is a measure of the accuracy of the system and for the current moment, the tolerance of deviation is set to 10% of traveled distance.

Table 1. Technical Specifications.

Item	Specifications
Computation Time	50 (*ms*)
Maximum Deviation	10% of Travelled Distance (*m*/*mm*)

4. Design Approach and Details

4.1 Design Approach

In this project, we leveraged iterative prototyping approach towards our project completion. We identified critical and major components that are required for the project first, and iteratively improve, modify each component in our design.

The carrying ability of the blimp limited the camera we could use. The onboard camera only could provide image stream with limited quality. Image with noise affects the integrity of trajectory estimated by the visual odometry, thus we need to build a noise handling module. The camera already adopted for the blimp is a fisheye analog camera with FOV of 120 degrees. Fisheye camera imposes huge lens distortion on the image stream and to restore original features, we need to perform camera calibration. To avoid impact on noise recognition in the noise handling module, the calibration and rectification module is designed after the noise handling module.

To localize the blimp and generate trajectory, a module that is responsible for calculating camera poses are required. To visualize, record, and compare the trajectory estimated against the ground truth. Modules for visualization and benchmarking are also designed for the entire system.

The entire system is designed as the pipeline shown in Figure 1.

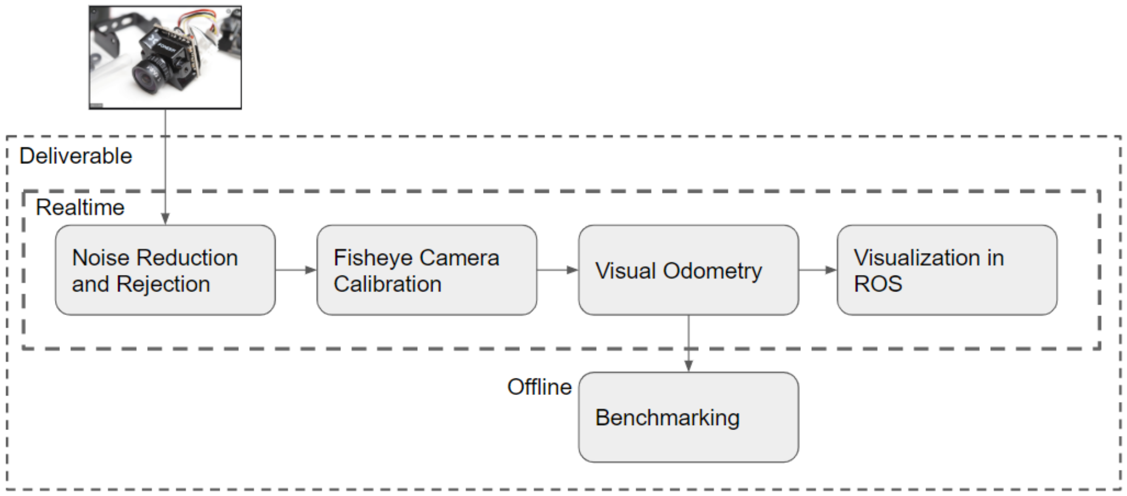


Figure 1. System-Level Block Diagram.

4.1.1 Noise Handling

The analog camera adopted in the blimp hardware design produce significant amount of noise as shown in Figure 2.

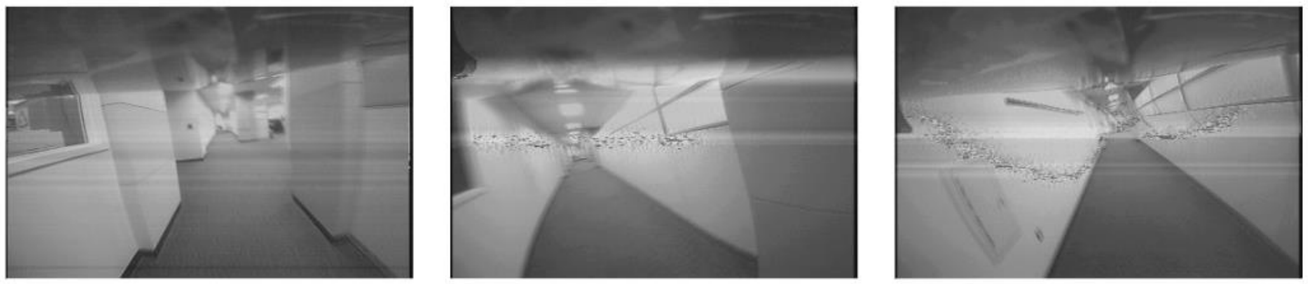


Figure 2. Example Noisy Image.

To reduce the influence of noise on the visual odometry estimation, we built modules for rejecting highly noisy images and repairing repairable images. For image rejection, we estimate variance of gaussian noise present in the image by performing a 1D Gaussian Kernel convolution vertically [8] and set up a threshold to filter images as shown in Figure 3.

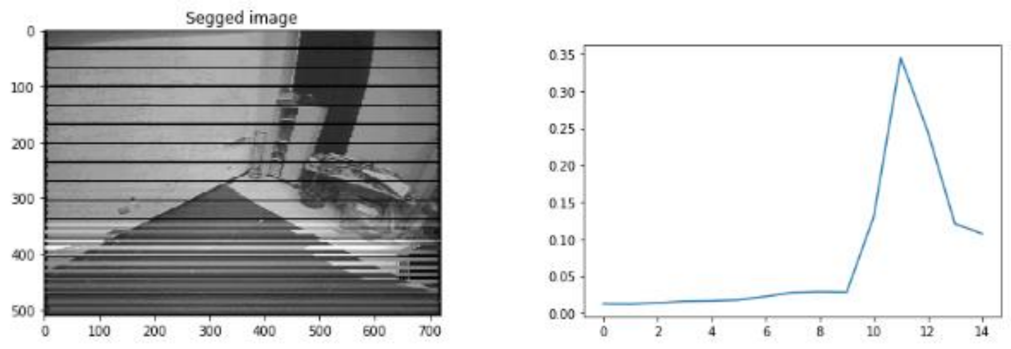


Figure 3. Image and Estimated Variance in each wide Row.

After rejecting unrepairable images, we filter low frequency noise in the Fourier domain and reconstruct the images, shown in Figure 4.

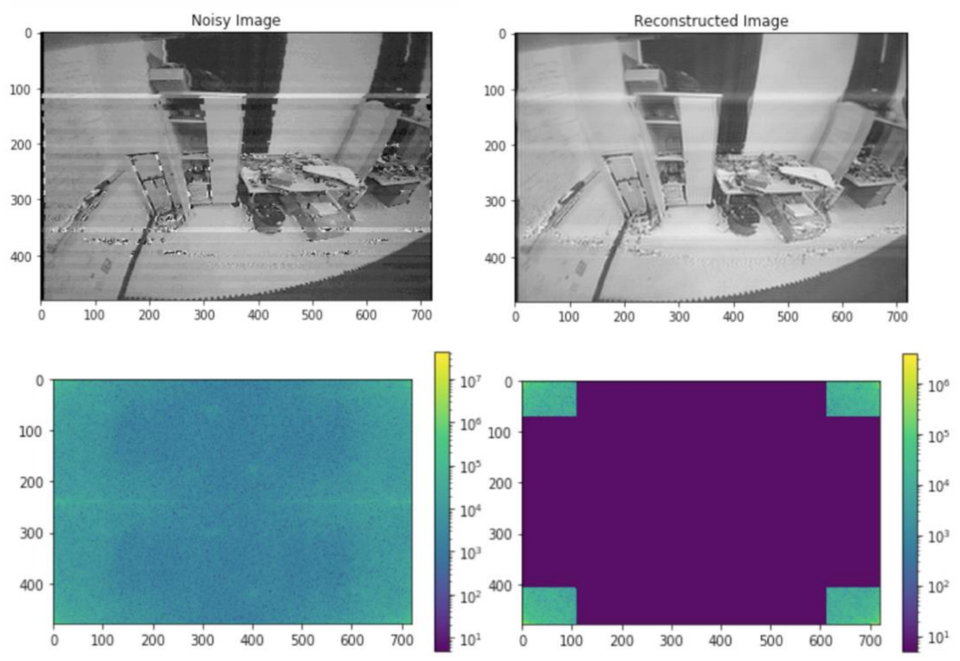


Figure 4. Denoise Noisy in the Fourier Domain.

4.1.2 Fisheye Calibration

Accurate visual odometry relies one assumption that any appears straight in real life also appears straight in image stream. The onboard analog camera is a fisheye camera with 120˚ field of view, which impose much distortion on the edge of the image, warping straight lines into parabola. We applied fisheye camera calibration via software package OCamCalib [9]. Due to image morphing, the eventual image requires to be further cropped and resulted in the image format shown in Figure 5.

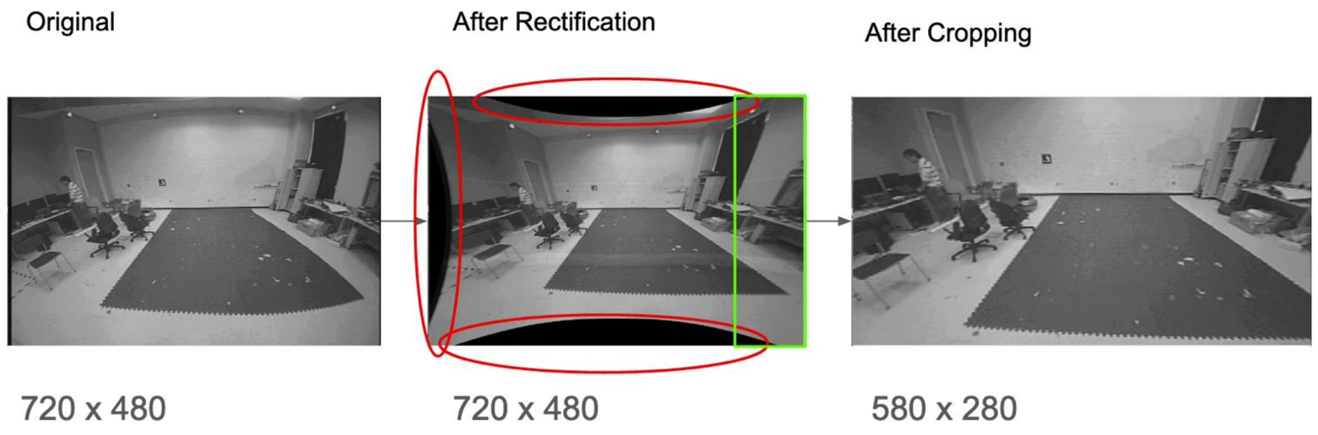


Figure 5. Fisheye calibration and Image Cropping.

4.1.3 Visual Odometry

Direct Sparse Odometry (DSO), an existing software package [7], was **modified** in this project to provide the odometry performance we require. DSO estimates camera pose change by minimizing the photometric error when projecting previous image onto the next one. DSO only selects a subset of pixels in the entire image for real-time computation effort. Selecting only a subset of pixels and no explicit feature matching between consecutive images provide unique merit in the noisy image stream of this project.

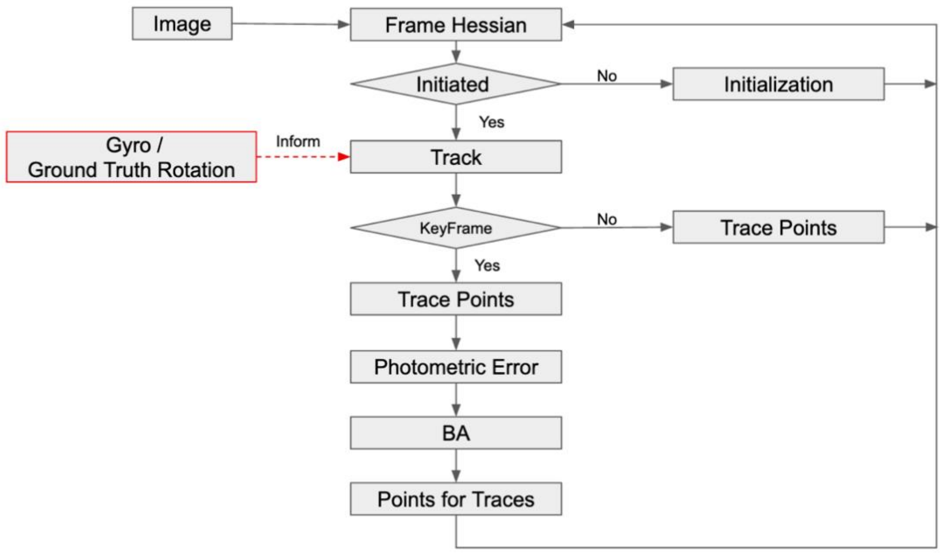


Figure 6. DSO with Modification Flow Chart.

Within the scope of visual odometry, Figure 6 shows DSO track frames, trace feature points, compute photometric errors and perform optimization to minimize photometric error to deduce the transformation between camera poses.

We modified the software package so that it additionally receives a ground truth orientation information (as marked in the red box) to help solving the optimization problem when there are a large rotation coupled with small translation, as such transformation would lose feature and/or present a poor triangulation problem for visual odometry.

4.1.4 Visualization

The visualization of the trajectory is simple and straight forward with Rviz provided by ROS.

However, OptiTrack orientation depends on the pose of the reflective ball and VO's initial direction and scale might be arbitrary.

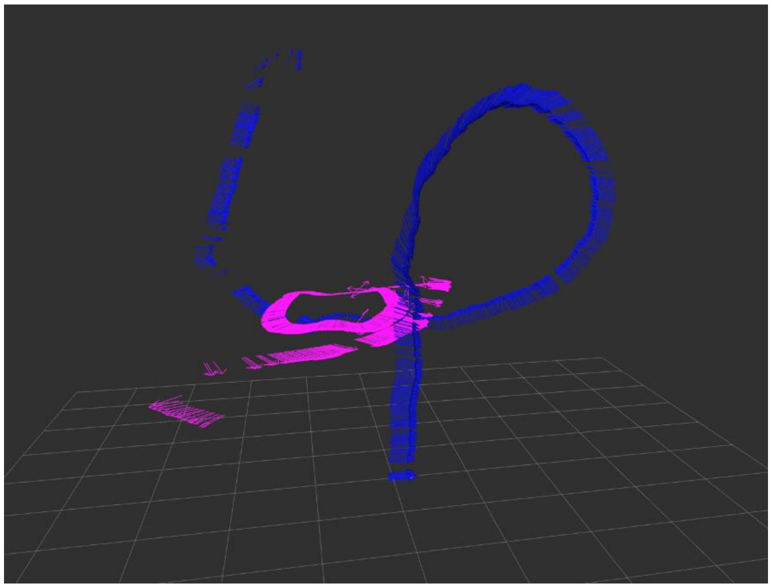


Figure 7. Untransformed Estimated Trajectory and Ground-truth Trajectory.

To make more intuitive visualization and benchmarking, we performed trajectory alignment. The problem is formulated as an optimization problem where a SIM (3) transformation with minimum MSE, is to be found. We took the first 200 points on the trajectory and the ground truth and performed such optimization, and the aligned trajectory is shown as Figure 8.

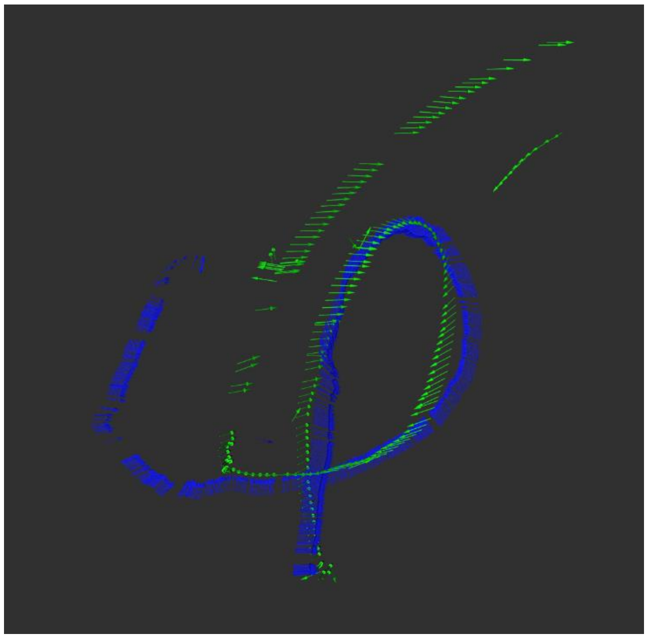


Figure 8. Transformed Estimated Trajectory and Ground-truth Trajectory.

The visualization is designed as shown in Figure 9. DSO publish ROS TF message into ROS system. A trajectory generator listens to the TF in ROS, generates trajectory, aligns it with the ground truth, and sends back to ROS. Rviz listens to the trajectory and perform the visualization itself.

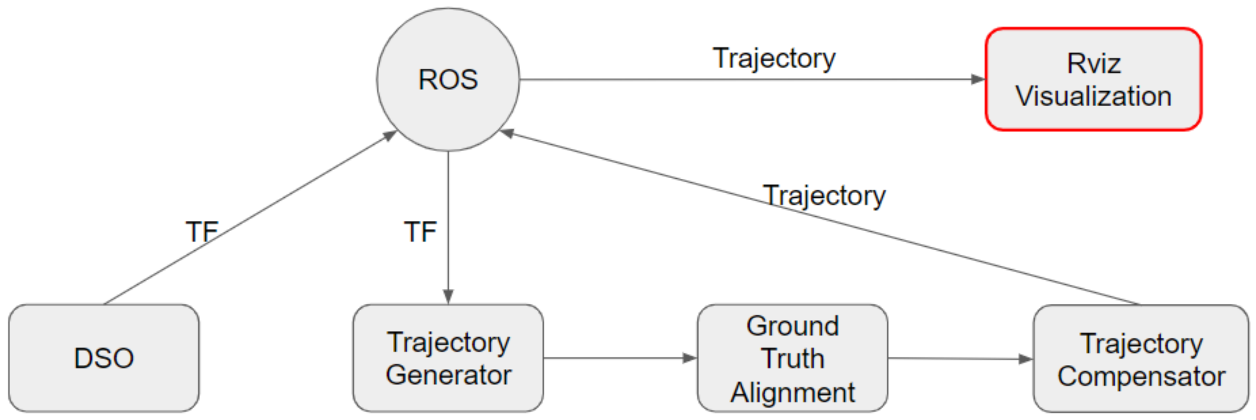
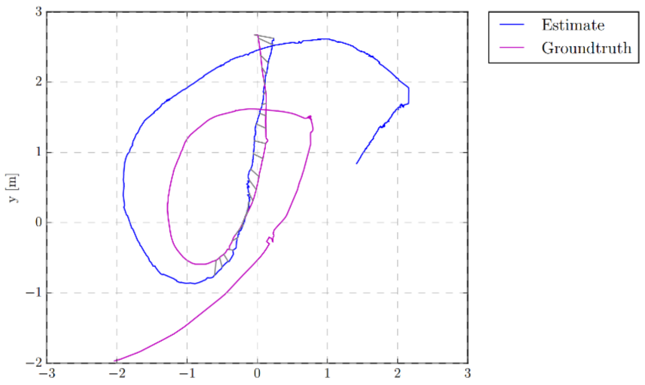


Figure 9. Visualization System-Level Block Diagram.

4.1.5 Benchmarking

To help evaluate the performance of the system and to find out which module require improvement, we also designed an off-line quantitative evaluation module based on RPG-Trajectory Evaluation [10] with slight modifications to adapt to our pipeline. The aligned trajectories are aligned and plotted in top view (x-y) and side view (x-z) axis to illustrate the difference between our estimation and the ground truth as shown in Figure 10.



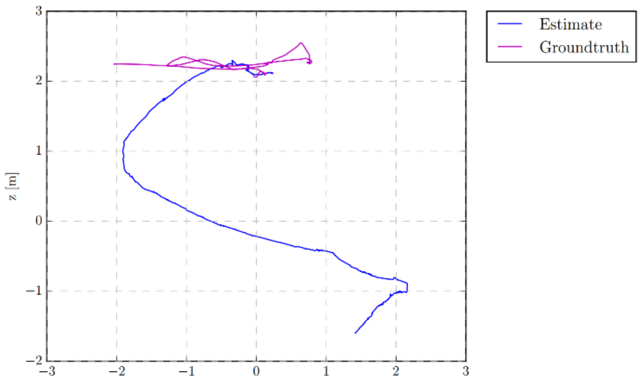
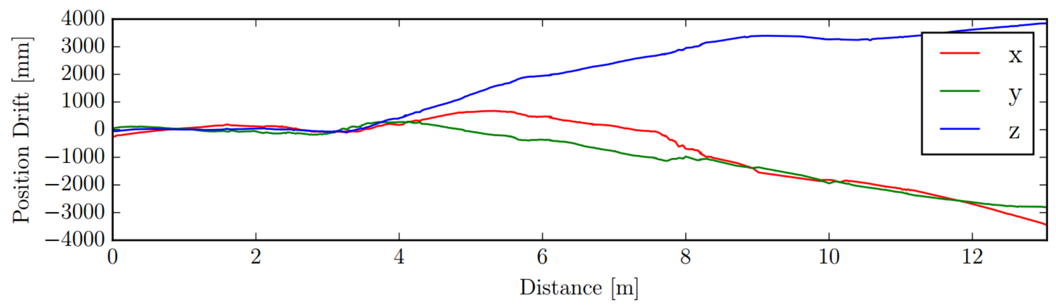


Figure 10. Top view (left) and side view (right) plot on comparison between ground truth and our estimation.

Drifts in x, y, z, axis, and scale are also quantitatively measured and plotted against distance travelled as shown in Figure 11.



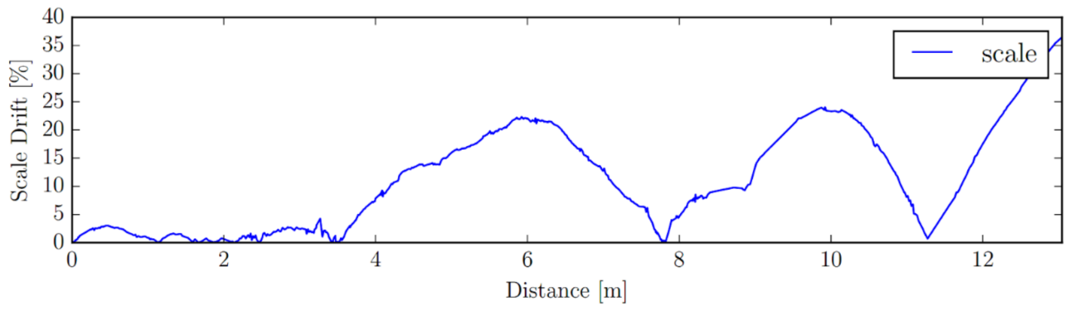


Figure 11. Drifts in x, y, z axis and drifts in scale against distance travelled.

Table 2. Computation Time of Visual Odometry

	Time (*ms*)
Average	3.448
Standard Deviation	2.334
Maximum	48.80

The computation time for the visual odometry was recorded via ROS's message filtering system and reported as in Table 2.

4.2 <u>Codes and Standards</u>

Since this is a software project building on a drone, the following code and standards apply to our project.

1. MIT Software License: Permissive free software license to put on software packages that impose minimal restrictions on reuse. This permits our design to be reused by most people on the internet yet restricted us from using certain proprietary packages such as GPL.
2. FAA Part 107: Drone regulation in public air space. UAVs (unmanned air vehicles) must fly under 120m and under 45m/s during the day. This regulation set the highest speed we expected the blimp to operate.
3. Inter-integrated Circuit (I2C) features a maximum clock frequency of 400 kHz and can be used to connect low-speed devices like microcontrollers. We follow the protocol to establish the connection between the controller and an IMU sensor.
4. Universal Serial Bus (USB) features a high speed of 480 Mbit/s and multiple peripheral devices connections.

These design standards, however, do not significantly or directly affect our design choices as our design is only on perception of the world and does not actively actuate any part.

4.3 Constraints, Alternatives, and Tradeoffs

Most of the constraints come from the existing hardware of the blimp. Such constraints limit our approach and methods in software system design and bring trade-offs in both qualitative and quantitative manners.

The blimp platform (GT-MAB) can only take 70 grams load maximum and the onboard controller itself weighs 40 grams. Low carrying capability of the platform limits the camera available onboard. Dedicated global shutter cameras that are widely adopted for visual SLAM and VO system have large sizes and heavy weights that are not feasible to be carried by the blimp. The only choices are analog and digital cameras that could only provide limited image resolution and significant amount of noise. The original design of the blimp adopted an analog camera module with prebuilt 5G transmission on it [1]. To keep minimum impact on the current hardware platform, we kept the original hardware settings though we tested the performance on digital camera with lower noise.

The hardware limitation of the camera brings qualitative tradeoff in designing software pipelines.

Popular visual odometry includes direct and indirect methods. Indirect methods leverage features such as SIFT, FAST, ORB and use bundle adjustment to look for transformation that minimize the matching errors. Direct methods sample all pixels and form optimization problem to minimize reprojection photometric error. With the limitation on the camera quality, we chose and only could chose direct methods, because features detectors in indirect methods would be disrupted by the image noises and generate random features that throw off the odometry optimization process [7].

Visual odometry algorithms are also divided in to sparse and dense approach. Sparse methods sample certain points and features form all the pixels while dense approach explores the information from every pixel in the image stream. Usually, dense methods require longer computation time. Considering about the real-time requirement, we are leaning towards sparse approach.

There is also quantitative trade-off in parameter tuning. Noisy images are rejected based on the estimated variance. Setting this threshold too high would cause extensive noisy image to compromise the integrity of visual odometry, while too low would keep the visual odometry system from receiving enough images to deduce the camera motion. Experiments were conducted to find the best threshold.

In the visual odometry, number of points tracked also involves trade-offs. Larger number of points tracked result finer optimization and more accurate estimation but sacrifice the computation speed. We performed experiments to look for parameters that enable real-time computation and result in the smallest drift.

5. Schedule, Tasks, and Milestones

Schedule, Tasks and Milestones are included as a Gantt chart in the Appendix A.

6. Final Project Demonstration

The project demonstration of this project involves demonstrating that the estimated visual odometry generates a sensible trajectory when compared to the ground truth trajectory. To demonstrate this ability, we provide the following documentation in Appendix B.

1. A video of side-by-side estimated trajectory and ground-truth trajectory generated in real time.

2. An Image of estimated and ground-truth trajectory.

To validate and evaluate our system. We use the mean-squared-error (MSE) between estimated and ground-truth trajectory versus traveled distance to show how much the odometry drifts from its target. This has been performed and showcased in section 4.1.5, as building this benchmarking package requires much effort.

The “ground-truth” trajectory referenced in the above section is provided by the OptiTrack system installed in the lab. The data was recorded so that each Image has a corresponding 3D Blimp pose which enabled pose-to-pose mean-squared-error analysis.

We met the technical specification of maximum deviation < 10% travelled distance up until 8m. Computation time of the visual odometry was also highlighted in section 4.1.5 and we meet the technical specification of 50ms.

7. Marketing and Cost Analysis

7.1 <u>Marketing Analysis</u>

With visual odometry, the blimp system now can localize itself in most of the condition without help from an external localization system. The benefits of the visual odometry enable more complicated and advanced applications to be built on top of this current system. Such applications include autonomous navigation, swarm control, and human robot interaction.

Additionally, the blimp platform is soft, safe, and have long flight length. These benefits combined with the capability of visual odometry, enable applications such as following: warehouse patrol, and indoor tour guide.

Specifically, with current situations the blimp can be applied to guiding patients around the COVID19 treatment and testing sites. There is not much existing similar product that could fly around with such safety considerations [1]. This project adds new feature to the novel platform and enables it with more functionalities.

7.2 Cost Analysis

The cost of building a blimp includes:

Table 3. Blimp Component Pricing

Component	Price (USD $)
Blimp	5
Arduino Controller	10
3D printed Gondola	10
Motors	10
Camera	20

Total costs of a blimp including helium gas and other miscellaneous items such as tape, wire, solder boost the cost of a blimp to around $100.

Our software stack design does not impose a cost on the already built blimp platform. However, it requires a computer that can run the algorithms, usually this implies a mid-end desktop or high-end laptop, which typically cost around $800.

8. Conclusion

Applying visual odometry on a mobile platform with limited hardware quality is challenging yet achievable. Only with extensive effort in both image processing and modification to state-of-the-art visual odometry techniques we were able to achieve the current level of estimated trajectory. The final measurements of specification come to MSE of near zero drift at 4m, averaged 1600mm drift at 8m and 3300mm drift at 13m.

Through this project, we learned that gaussian assumption is often a good start for any random noises. Any machine learning algorithms would require a good conversion technique to convert any type of data into easily learnable data formats.

Visual odometry would fail when large rotation coupled with tiny translation, either via loss of feature of optimization unable to find a global minimum. Warm starting the optimization through a rotation that is close to the solution would help solve this issue.

There are some of options to extend our work, one would be the full integration of an IMU to compensate the scale problem in pure visual odometry, and another would be to use this visual odometry to conduct map building.

9. Leadership Roles

According to the specialty of each team member, different leadership roles are assigned as in Table 4.

Table 4. Leadership roles.

Fanzhe Lyu	1. Software Lead 2. Testing Lead
Ruoyang Xu	1. Director of Communications 2. Hardware Lead
Yifan Shen	1. Documentation Coordinator 2. Expo Coordinator 3. Webmaster
Yilun Xie	1. Design Lead 2. Testing Lead

10. References

[1] F. Zhang, “I-Corps: Autonomous Indoor Blimps as Data Terminal for Internet of Things,” *NSF Award Search: Award#1824233*, 09-Mar-2018. [Online]. Available: https://nsf.gov/awardsearch/showAward?AWD_ID=1824233&HistoricalAwards=false. [Accessed: 25-Apr-2020].

[2] Natural Point Inc., “Motion Capture Systems,” *OptiTrack*. [Online]. Available: https://optitrack.com/. [Accessed: 25-Apr-2020].

[3] Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. (2002). "FastSLAM: A factored solution to the simultaneous localization and mapping problem" *(PDF)*. Proceedings of the AAAI National Conference on Artificial Intelligence. pp. 593–598.

[4] R. Mur-Artal, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," in IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, Oct. 2015.

[5] C. Forster, M. Pizzoli and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, 2014.

[6] M. Bloesch, S. Omari, M. Hutter, R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach", Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots Systems, 2015.

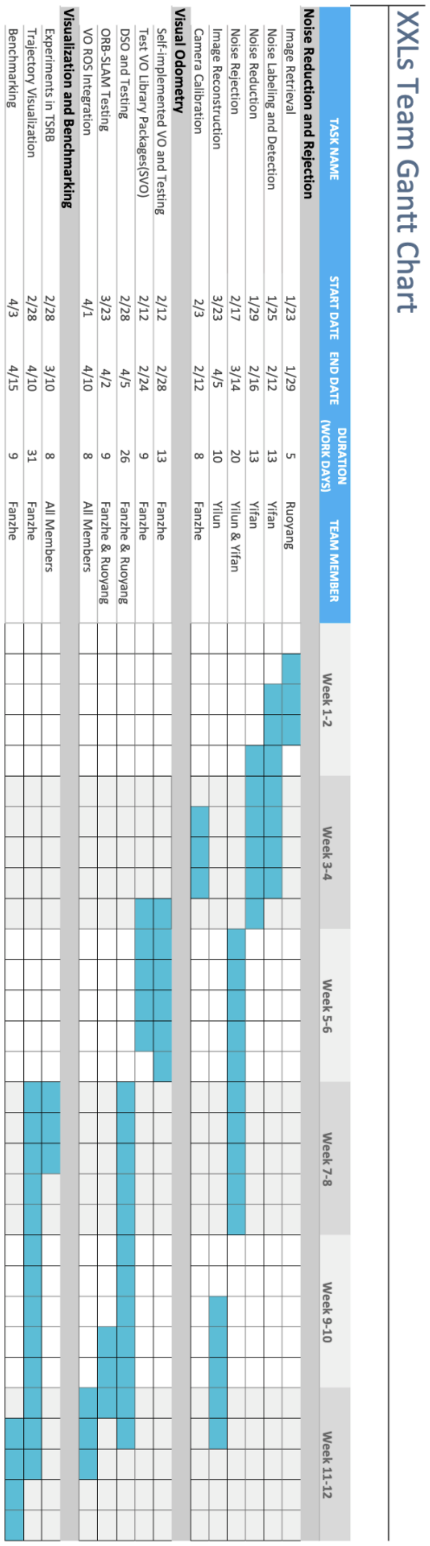
[7] J. Engel, V. Koltun and D. Cremers, "Direct Sparse Odometry," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 3, pp. 611-625, 1 March 2018.

[8] K. Rank, M. Lendl, and R. Unbehauen, "Estimation of image noise variance," in IEEE Proceedings – Vision, Image and Signal Processing, vol. 146, no. 2, pp. 80-84, April 1999.

[9] Scaramuzza, D., Martinelli, A. and Siegwart, R. "A Toolbox for Easy Calibrating Omnidirectional Cameras", Proceedings to IEEE International Conference on Intelligent Robots and Systems (IROS 2006), Beijing China, October 7-15, 2006.

[10] Z. Zhang and D. Scaramuzza, "A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018, pp. 7244-7251.

Appendix A: Project Gantt Chart



XXLs Team Gantt Chart

TASK NAME	START DATE	END DATE	DURATION (WORK DAYS)	TEAM MEMBER
Noise Reduction and Rejection				
Image Retrieval	1/23	1/29	5	Ruoyang
Noise Labeling and Detection	1/25	2/12	13	Yifan
Noise Reduction	1/29	2/16	13	Yifan
Noise Rejection	2/17	3/14	20	Yilun & Yifan
Image Reconstruction	3/23	4/5	10	Yilun
Camera Calibration	2/3	2/12	8	Fanzhe
Visual Odometry				
Self-implemented VO and Testing	2/12	2/28	13	Fanzhe
Test VO Library Packages(SVO)	2/12	2/24	9	Fanzhe
DSO and Testing	2/28	4/5	26	Fanzhe & Ruoyang
ORB-SLAM Testing	3/23	4/2	9	Fanzhe & Ruoyang
VO ROS Integration	4/1	4/10	8	All Members
Visualization and Benchmarking				
Experiments in TSRB	2/28	3/10	8	All Members
Trajectory Visualization	2/28	4/10	31	Fanzhe
Benchmarking	4/3	4/15	9	Fanzhe

Appendix B: Project Demonstration

We hereby link to a page in our website that contains a video of the estimated visual odometry compared against the ground truth trajectory. Link: http://ece4012y202002.ece.gatech.edu/sd20p37/

Appendix C: Source Code

All our source code exists in private repositories of the organization *Senior-Design-XXLs* in the GitHub hosted by Georgia Tech. Currently only our advisor and the hosting laboratory has access to the code. To request access to the code, please email the project advisors listed on the front page.

Link: https://github.gatech.edu/Senior-Design-XXLs