Deep Monocular Visual Odometry For Robotic Blimps

ECE4011 Senior Design Project

XXLs (Team 30)

Advisor: Dr. Fumin Zhang

Fanzhe Lyu	CmpE	fanzhe@gatech.edu
Ruoyang Xu	CmpE	rxu74@gatech.edu
Yifan Shen	CmpE	aelitashen@gatech.edu
Yilun Xie	CmpE	michaelxie.yl@gatech.edu

Submitted

11/18/2019

Table of Contents

Executive Summary

An existing robotic blimp (GT-MAB) platform was designed to offer long flight duration, quiet sound profile, and safety for collisions. However, in order to execute sensing tasks successfully, odometry estimation needs to be performed onboard so that the blimp can know its location. The proposed project is modifying hardware and expanding perception capacity for the robotic blimps by building a visual odometry system, using a monocular camera to estimate odometry through its movements. The proposed project will potentially unleash SLAM capacity for the blimp. Moreover, spatial relationships on objects sensed could be established and algorithms such as automatic navigation can be built upon.

The modified hardware system will include the blimp itself, an onboard camera and an IMU sensor. A separate workstation will support the robotic blimp by processing sensor data transmitted and issuing control commands via 5.8GHz transceivers. Deep learning techniques will be used to estimate odometry from visual inputs and IMU data. DNN model will be trained using data provided by a motion tracking system and should be able to apply on sensor data in any unknown environments to generate an estimation of robot odometry. The system will be prototyped in a virtual environment with Gazebo and ROS, or similar software where virtual data can be generated. The performance of the system will be measured by comparing the estimated odometry calculated by the system and provided by ground truth supported by simulation software (prototype) and a motion tracking system (real world). Less difference in odometry between estimation and ground truth will indicate better system performance.

The expected outcome will be a blimp with an attached camera and IMU sensor with the capacity of knowing its own location during any flights through visual odometry software. The building costs mainly from hardware mentioned above and will be at $200. With anticipated replacements for malfunction parts caused by possible control failures of the aerial robot, a total estimation will be around $400.

Nomenclature:

Acronyms	**Definition**
CNN	**Convolutional Neural Networks**
DNN	**Deep Neural Networks**
FAA	**Federal Aviation Administration**
GHz	**Gigahertz / 10^9 Hertz**
GPL	**Gnu-Permissive License**
GT-MAB	**Georgia Tech Miniature Aerial Blimp**
I2C	**Inter-Integrated Circuit**
IMU	**Inertial Measurement Unit**
kHz	**Kilohertz / 10^3 Hertz**
LSTM	**Long-Short Term Memory**
NTSC	**National Television System Committee**
PCB	**Printed Circuit Board**
ROS	**Robot Operating System**
SLAM	**Simultaneous Localization and Mapping**
UAV	**Unmanned Aerial Vehicle**
VO	**Visual Odometry**

Deep Monocular Visual Odometry For Robotic Blimps

1. Introduction

The project is building visual odometry algorithms for robotic blimps to estimate the location and direction through a monocular camera and associate IMU data. The system consists mainly of a robotic blimp, an IMU sensor, a monocular camera, and a 5.8GHz transceiver system. The system will cost around $200 in total, and with probable replacement costs for malfunction parts caused by possible control failures of the aerial robot, the team is requesting $400 to develop the system.

1.1 Objective

The objective of the project is to modify existing blimp platform hardware and to create a deep learning visual odometry algorithm for robotics blimps to estimate location and direction through images captured by a monocular camera and data collected by an IMU sensor. The mounted onboard camera and IMU sensor will provide the system with a continuous stream of images and IMU data. A deep learning model will be created and trained using both virtual and real data with ground truth provided by simulation software and a motion tracking system. The trained model will take in images and IMU data and provide the robot with knowledge of the trajectory traveled by the robot, as well as the location and direction of the robot itself.

The expected system will achieve high precision localization capability with only a camera and an IMU sensor and without any information provided by external resources such as motion tracking systems. The performance of the visual odometry system will be evaluated by comparing the odometry estimation with ground truth captured by motion tracking systems. Less difference will be an indication of a better-performed system.

1.2 Motivation

Persistent environmental monitoring, infrastructure inspection, and security surveillance are essential for large indoor facilities to provide safety and comfort. Existing aerial sensing solutions are mainly for outdoor applications. They are not suitable for human-occupied indoor environments due to short flight duration, disturbing noise level, and safety concerns. Existing robotic blimps (GT-MAB) are designed to offer long flight duration, quiet sound profile, and safety for collisions. The technology can potentially fill the market vacancy for indoor aerial sensing, as well as research and education purposes.

However, in order to establish spatial relationships of objects sensed and execute sensing tasks successfully, odometry estimation needs to be performed on board so that the blimp can know the trajectory it traveled and its location. Due to the high costs and inconvenience of building motion tracking systems in large areas such as datacenter and warehouses, using visual sensors to estimate robots' odometry is a more affordable and accessible solution. The motivation of the project is to build an odometry estimation and localization system by applying deep learning on camera inputs and IMU data. With visual odometry estimation, mapping and localization for the blimp can be achieved and autonomous navigation algorithms can be built upon, in an affordable and accessible fashion.

The development of the visual odometry algorithm is critical for blimps to achieve autonomous indoor sensing features such as tools for indoor map building. With a combination of other software such as autonomous navigation algorithms, usages of blimps can be extended to infrastructure inspection and security surveillance for large indoor areas such as warehouses and data centers.

1.3 Background

Visual odometry is critical for robots to perform a perception task because localization is always essential for associating spatial relationships. Due to the software nature of visual odometry and its reliance on fast hardware, until the early 2010s, application of visual odometry primarily exists in the fields of academia and pure research, with the application on martian rover being the most prominent [1]. Recent application utilizing visual odometry orients around the autonomous car and drone companies. Skydio uses 13 cameras to perform its visual-based simultaneous localization and mapping task, which is a combination of visual odometry and localization tasks and puts the price of the entire drone at $1999 [2]. A competing company of Skydio, DJI, announced a hardware platform for visual odometry platform research priced at $1249 [3, 4] in 2015.

Deep learning has proved its power in processing visual information. In tasks such as object segmentation and image processing has driven research on applying deep learning on visual odometry. Research on deep visual odometry with monocular cameras mainly focused on technical issues of the spatial ratio. Correct spatial ratios are difficult to be obtained from a monocular camera due to hardware constraints. Recent research combining visual inputs and IMU data showed promising results.

On the other hand, despite the availability of hardware platforms for deep and conventional visual odometry algorithms, odometry methods rarely exist as a standalone ready-to-ship product. The majority of design comes as a proprietary part of a bigger product such as those for Skydio and DJI, and other designs are released as open-source projects aimed at promoting research in the field. SfMLearner is a deep visual odometry project open-sourced on GitHub done by Google. This work features in its ability to conduct unsupervised deep learning and generating depth data with a single-view camera [5]. OpenVSLAM is another open-source

GitHub project for building visual odometry research with fully modular design and compatibility across multiple camera models as its primary feature [6].

1.4 Proposal Organization

The proposal is organized to first cover project descriptions, requirements and specifications (Sect. 2), followed by engineering specifications (Sect. 3). Section 4 focuses mainly on technical approaches in the design as well as analyzation of the decision chosen. Project demonstration (Sect. 5), project milestones and timeframe (Sect .6), marketing and costs analysis (Sect .7) will then be covered. Current status and leadership roles will be explained at the last.

2. Project Description, Customer Requirements, and Goals

The appropriate visual odometry algorithm for GT-MAB will be designed and prototyped by the team. The visual odometry algorithm would be effectively and efficiently estimating the location and direction of the robotic blimp based on the image captured by the monocular camera and the data collected by the IMU. The odometry of the whole system will be updated properly as well. The critical design of the project is implemented in three steps: stable image acquisition, neural network implementation and odometry updates.

Targeted users of the project would be individual users that needs low-cost indoor monitoring devices, laboratory researchers who carry out researches focus on VO algorithms and DNN and industry companies that make drones and blimps. The stakeholders of this project include individual customers, industrial manufacturers and school labs for either educational or research use. More details are listed in Stakeholders 2x2 table below:

Table 1. Stakeholders 2x2 Table

Influence (Low - High)	Individual Customer for indoor aerial sensing	Drone companies, Companies of indoor monitoring systems
	Blimps companies, Companies of specialized DNN chips	College Labs involving VO research computer vision and DNN
	Interest (Low - High)	

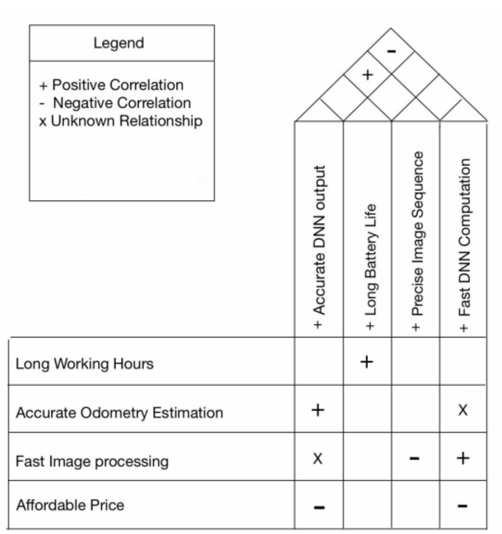
The customer requirements of this project are listed below:

1. The blimp is able to work for long hours and will not be destroyed easily.
2. The estimation of location and direction is consistent and instant.
3. The odometry is reliable and precise.
4. The price of the whole system is affordable.

The final product should offer reliable and precise estimation of the location and direction of the blimp. Within the long hours that the blimp keeps capturing the information from indoor environment, the odometry should be accurate without external resources. The target price for the product is about $200 and that of the replacement units is below $100.

The system is expected to continuously collect visual information, i.e. images of the surroundings through the monocular camera equipped and data from the IMU sensor attached. The system needs to be working ideally more than 8 hours after every full charge. The system will utilize data provided by a motion tracking system to train the DNN. When the blimp is wandering around an unknown indoor environment and keeps collecting essential information, the system needs to preprocess the images from frame to frame to offer stable inputs for the neural network. The DNN is responsible for updating the odometry and providing an estimation of the environment with more than 90% accuracy based on the information fed in a high frequency. Accordingly, associated engineering performance metrics include long-hour working time (more than 8 hours), accurate and precise odometry estimation, reliable camera devices and IMU sensor. A QFD chart that illustrates customer needs and engineering specification is below:

Table 2. QFD Specializing Customer Needs and Engineering Requirements



The blimps are very lightweight and can only carry a limited sensor. Currently the blimps only carries a monocular camera and RGB is the only source of odometry estimation, which induces further problems: visual odometry misses the scale from the image capturing process thus requires extra calibration of the images, otherwise it is unrealistic to expect the blimp to work properly in an unknown environment. The current system needs to be outfitted an extra IMU to produce scale information.

To sum up, the features of the system include:

1. smooth movement of blimp upon remote control
2. real-time analysis of image and sensor data through DNN
3. reliable connection through I2C between IMU and main controller
4. enduring operation hours and add-on devices

3. Technical Specifications

The two primary specifications for the project would be estimator inference time and maximum deviation distance. Table 3 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 inference 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 3** contains relevant specifications for the design.

Table 3. Technical Specifications for the Design

Item	Specifications
Inference Time	50ms
Maximum Deviation	10% of traveled distanced

4. Design Approach and Details

4.1 Design Concept Ideation, Constraints, Alternatives, and Tradeoffs

Ideation

The objective of this project is to use consecutive monocular images and IMU data to obtain any translational information to update the blimp's odometry, through a trained neural network. Thus, there are three main functions needed to be implemented: stable images acquisition, neural network implementation and odometry update.

In compensation with the non-linear dynamics of the blimp during operation, the images captured by the onboard camera should undergo correction that would stabilize the image from frame to frame. Training the neural network requires training and testing data and they can be created by associating monocular camera inputs and IMU data with ground-truth odometry captured by motion capturing system. The largest portion of work, however, lies within training and fine-tuning the neural networks. The choice of the number of hidden layers and the number of nodes in the hidden layers will all potentially affect the accuracy and the training time of the neural network. The more hidden layers we use, the more capability for fitting input data, resulting in potential increase in accuracy but at the same time making the neural network training much harder and time consuming.

Alternatives

The majority of design comes as a proprietary part of a bigger product such as those for Skydio and DJI, and other designs are released as open-source projects aimed at promoting research in the field. SfMLearner is a deep visual odometry project open-sourced on GitHub done by Google. This work features in its ability to conduct unsupervised deep learning and generating depth data with a single-view camera [5]. OpenVSLAM is another open-source GitHub project for building visual odometry research with fully modular design and compatibility across multiple camera models as its primary feature [6].

It is also possible to train the neural network using the data that directly map their input to the positional information. In that case, the process of updating the transitional data to the odometry is eliminated, making the entire computation more efficient. However, this change is likely to make training more challenging. More hidden layers are expected to be added to compensate for the increasing nonlinearity between the input data and the output data. This option will also be more error-prone by subjecting all the error to the performance of the neural network. Thus, obtaining the most accurate neural network is vital to making this option work.

Constraints

One significant constraint for this project is that the extremely light-weighted body of the aircraft, making it susceptible to disturbance and not suitable for outdoor applications. Thus, the blimp is limited to indoor application scenarios. The light-weighted body also puts constraints over its carrying capacity: heavy sensor like radar is unable to be mounted on the aircraft; camera and IMU are the only two sensors that we can rely on.

Another constraint is due to the very nature of visual odometry: the scale is missing from the image capturing process. So, either we need a precomputed map of the indoor environment or we need a fiducial object (an object with known dimensions) to calibrate the image into a known scale. Putting the aircraft into an unknown environment and expect it to work is simply unrealistic in this case. The missing scale is the reason why we integrate an IMU into the system, and the details of this implementation will be further discussed below.

Trade-offs

One main trade-off is the quality of the image transfer and the real-time processing capacity of the system. Transmission of the camera-captured images are first delivered to the radio base through serial communication and then transferred to the lab computers via wireless transmission. The network congestion can cause evident delays in our real-time system. A series of convolutional neural networks can concurrently process multiple images while the down-sampled output can affect the image quality. The second trade-off is the mobility and stability of the blimps. Setting a higher moving speed can traverse the environment more efficiently, but the blimp is less likely to be stable, which would hamper high-quality image gathering.

This project is targeting at high updating frequency for the real-time. Therefore, the speed of the network would also be one of the essential design considerations. Because image processing and neural networks inference are performed on a ground workstation, dedicated computation hardware such as GPU can be used to shorten the calculation time.

4.2 Preliminary Concept Selection and Justification

Selection Criteria

The objective is defined to be visual odometry methods that utilize deep learning methods. Considering the constraint of our hardware platform, which is a lightweight blimp that only houses a single camera onboard, the selection criteria for the design have the focus on “monocular” and deep visual odometry methods. While more than several visual odometry methods came as promising candidates: [7] and [8], DeepVO [9] stands out in its end-to-end elegance and a compact structure for modification as the core algorithm for our design.

System Overview

The implementation of the deep visual odometry estimator involves a software implementation of the algorithm and the hardware platform that collects the ground truth training data as well as the testing platform. Figure 1 is a block diagram that visualizes the conjugation of the two systems.

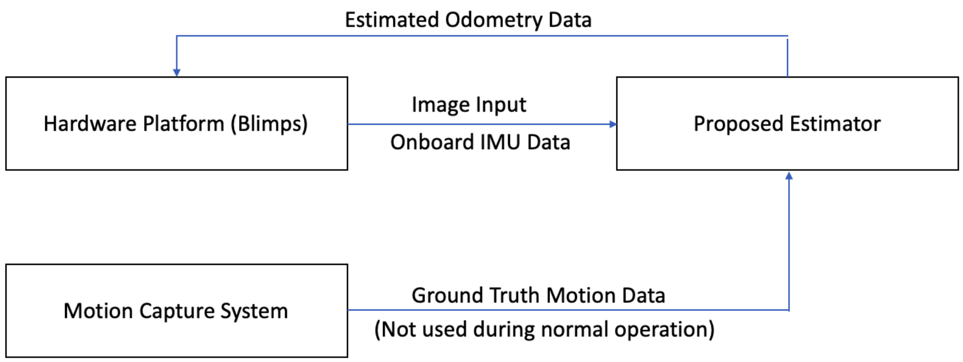


Figure 1. Software Implementation Overview.

The motion capture system, with the modified blimp platform, will provide the necessary ground truth sensor data for the software implementation. The software implementation would utilize the ground truth data to train a neural network estimator. The effectiveness of the application will be evaluated in a simulated environment before moving to the hardware platform. This virtual environment will be discussed in the analyses and experiment section.

Software Design

Due to the constraint present in monocular visual odometry, an estimator that only relies on camera inputs is likely to generate much noise and very unreliable. On top of the visual estimator, we would utilize an onboard IMU to help create more reliable output, and the design is thus split into two components:

The first core component of our deep visual odometry estimator is the method described in [9] published in 2017. The paper proposed an end-to-end deep neural network for monocular visual odometry named DeepVO. DeepVO has the architecture of recurrent convolutional neural networks (RCNN). Figure 2 shows the proposed architecture of DeepVO. At each time step, the two consecutive images are stacked together so that DeepVO can learn how image changes correspond to motion information. Conceptually, the convolution part of the network would transform the input images into a different feature space as feature vectors, and the recurrent section of the system would attempt to locate the temporal relation between feature vectors to generate an output of pose.

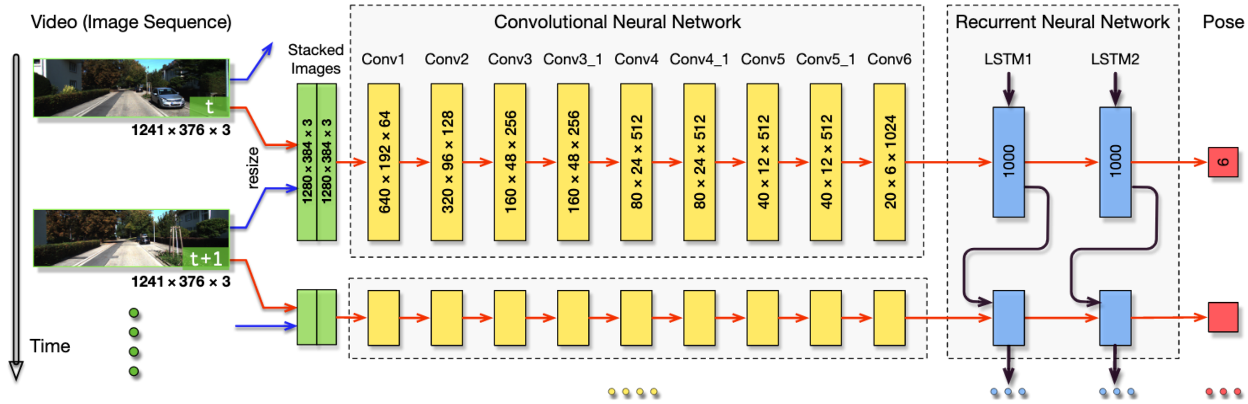


Figure 2. Neural network architecture of Deep VO.

Consider the fact that we are utilizing monocular visual odometry, there lacks a proper scale transformation between each consecutive image input. This would potentially result in overall offset of generated odometry. To compensate for this issue, an IMU would be utilized in conjunction with the deep visual odometry estimator. IMU generates the motion information and while it being a semi-reliable input source of odometry, IMU often suffers from inaccuracy and offset error overtime. Having an IMU input to the network would solve the problem of scale variant in pure monocular deep visual odometry and resolve the problem of unverifiable noisy IMU sensor data. The selected IMU module would be BNO055 and is within the weight limit of the current platform.

In summary, the software implementation would be integrating the output of a monocular deep visual odometry and the onboard IMU sensor data to generate a reliable odometry message. Figure 3 is a summarization of the software implementation.

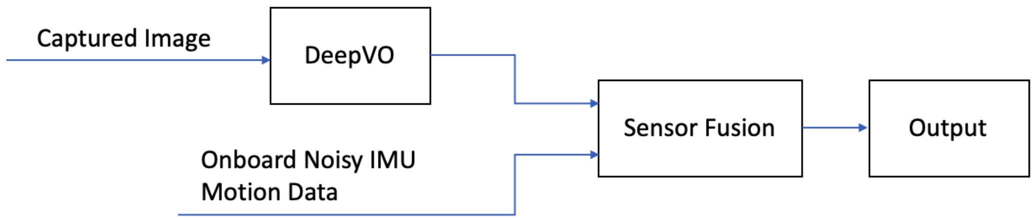


Figure 3. Odometry estimator overview.

It is worth noting that the team currently does not have a verified method for combining the motion information with the two different sources. This is still an ongoing investigation, and there are multiple paths available. One would be a traditional sensor fusion approach from the aspect of the adaptive filter, and the other would be a deep learning approach. The design team is uncertain which method would be the best approach, and we plan to adapt our approach as the implementation goes on.

Implementation Details

Due to the plentiful availability of machine learning toolboxes in Python, a significant portion of the software implementation would be done in Python, utilizing Tensorflow. The critical path lies on the DeepVO since the entire implementation is very linear except for the part where we wish to obtain two sources of motion information. Motion information from IMU is expected to be relatively straightforward, and therefore DeepVO implementation is determined to be a critical component of the design.

If the DeepVO fails to yield consistent results, we will utilize other visual odometry methods. They would be selected from the poll of unselected ways that were mentioned earlier in section 4.1.

Infrastructure and Hardware Platform

The odometry estimator would be applied on a blimps system currently developed at Georgia Tech Systems Research Lab. The blimp is an unmanned aerial vehicle that is designed to be interaction friendly. Figure 4 is an image of the blimps. It uses helium to provide the majority of atmospheric buoyancy. An Arduino Fio controls it with a custom PCB. 3 TB6612FNG drivers are used to driving 5 Qx Blade Pic motors. It houses a 2.4 GHz NTSC camera and interfaces with the rest of the system through radio. The current blimp does not contain an IMU and BNO055 is the intended IMU addition to the system.

Figure 4. Blimps platform from bottom viewpoint

Currently the lab has a motion capture system for the blimps to acquired ground truth odometry data. This will be the primary source of training data into the deep visual odometry estimator.

4.3 Engineering Analyses and Experiment

The visual odometry estimator is expected to be evaluated in a simulated environment before applying to the hardware platform. Gazebo is chosen as the simulated environment. Gazebo is a physics simulation toolbox widely used in the robotics community. It is chosen for its accurate modeling of the world, sensor noises as well as its infrastructure around ROS (Robot Operating System). ROS is another widely used robotic framework for interfacing with robots.

The design would be run in the simulated environment with simulated sensor inputs. Figure 5 is an illustration of the test environment. The primary concern of the design would be its deviation from true odometry. Since the algorithm would be run in a simulated world, access to ground-truth value is available. The output of implemented odometry estimator and ground truth value would be compared and evaluated first. The design of the system would be reiterated based on the result.

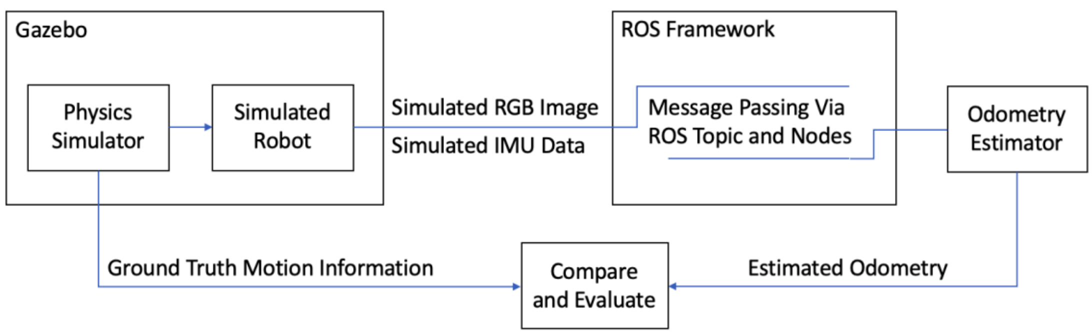


Figure 5. Simulation and experimentation environment overview.

Should the deviation remained in a reasonable tolerance, the experiments would proceed to benchmark the inference time of the estimator and simplify the model if necessary. The maximum speed usually determines the inference time limit the vehicle can travel. Since the blimps travel at about 0.5m/s, a target odometry update frequency would be 20Hz so that the maximum distance between each update would be 2.5cm. This would set the inference target to be at 50ms. The design would prune out unnecessary sections of the network to improve its inference time until it meets the standard.

4.4 Codes and Standards

1. MIT Software License: Permissive free software license to put on software packages that imposes 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 Intel MKL library with GPL license.
2. FAA Part 107: Drone regulation in public air space. UAVs must fly under 120m and under 45m/s during the day. This regulation does not explicitly regulate this project since the blimps are not capable of outdoor operation, it provides a guideline for the speed of recreational drone operation and a guideline for the target frame rate of the network output rate we need to achieve.
3. Inter-Integrated Circuit (I2C) is a serial protocol for two-wire interface to connect IMU and the main controller on the blimp. It features a maximum clock frequency of 400 kHZ and can be used to connect low-speed devices like microcontrollers, EEPROMs, A/D and D/A converters, I/O interfaces and other similar peripherals in embedded systems [10].
4. Universal Serial Bus (USB) is used to connect the Xbox controller and the PC. It features a high speed of 480 Mbit/s and multiple peripheral devices connections [11].

5. Project Demonstration

The demonstration of the algorithm will take place at Georgia Tech. To ensure the capability of demonstrating the performance of the system, the system needs to be demonstrated in the lab at Tech Square Research Building where the motion tracking system is installed. The deep visual odometry system should be demonstrated through the real-time odometry estimation of a blimp using camera and IMU data. The formal demonstration will consist of the following:

1. The user places the blimp in an arbitrary starting position and use the motion tracking system to register the position and orientation of the robot as the origin.

2. The user uses a joystick controller to fly the blimp to a sequence of arbitrary locations and use the motion tracking system to collect associate ground-truth odometry of the blimp.
3. Compare the odometry estimated by the deep learning algorithm with the odometry captured by the motion tracking system.

In order to demonstrate the specification, a PC and a monitor is needed to show real-time data output. The specifications can be demonstrated through the following:

1. **Visual Odometry Output:** The trajectory estimated using a deep visual odometry system will be plot in real-time through MATLAB.
2. **Deep Learning Inference Time:** Time consumed for each inference of a pair of input of the camera and IMU data will be measured in the software and be printed in the control console.
3. **Visual Odometry Error/Drift:** Mean absolute square error of the estimated visual odometry will be calculated and demonstrated in real-time.

6. Schedule, Tasks, and Milestones:

The visual odometry system will be implementing the prototype over the next three months. Appendix A contains the Gantt chart with time of major tasks and milestones. The Gantt chart delineated the associated start and end date of each tasks. Appendix B is a skill matrix and detailed the responsibility over tasks depending on the skills required by the task.

7. Marketing and Cost Analysis

Marketing Analysis is omitted due to this is an ECE senior design project.

7.1 Cost Analysis

The implementation cost for the software would none due to the fact that all software packages are open source and available as free. The training data will be acquired by the design team through experimentation. This would also be free as the motion capture system is already available in the lab.

The equipment cost would come as a large portion of cost and is outlined in **Table 4** and the development cost for the blimp is outlined in **Table 5**.

Table 4. Equipment Cost of the System

Product Description	Quantity per Blimp	Unit Price ($)	Total Price ($)
Arduino Fio	1	29.65	26.95
Custom PCB and Components	1	40 (Estimated)	40
TB6612FNG Driver	3	4.95	14.85
Qx Blade Pic Motor	5	9.99	49.95
NTSc Camera	1	7.99	7.99
BNO055 IMU	1	9.95	9.95
Helium Gas	1 (Liter with Tank)	45.25	45.25
Total Cost / Blimp			194.94

Table 5. Development Cost of the System

Description	Labor Hour (hr)	Labor Cost ($)
Visual Odometry Estimator		
Training data acquisition	10	400
Neural Network Implementation	40	1600
IMU installation and acquisition	40	1600
Sensor fusion	40	1600
Simulated Testing		
Construction of Simulation Environment	50	2000
Adaption of algorithm into the simulated environment	60	2400
Verification on Hardware		
Adaptation of algorithm to blimps platform	50	2000
Analysis of experiment outcome	20	800
Revision of algorithms	60	2400
TOTAL	370	14,800

The total development cost would be around $15,000. Since the design is a software product, instead of production runs that produces individual physical components, the design could be sold as subscription services that charges customers annually. To set the price, we consider the price of fringe benefit (30%), sales expense (8%), and technical support for the software package ($20/hr and approximately each sale requires 5 hour of support). Table X outlines the Assuming 1000 copies of software are sold over 5-year period, the total cost would go to $15000 \times (1 + 30\% + 8\%) + 1000 \times 5 \times 20 = \$120700$. To achieve a profit of $\$50$ per unit, the price would be set at $120700/1000 \ + \ 50 \approx \$170$

8. Current Status

Currently, the team has mainly in contact with a PhD student (Tony Lin) of the faculty advisor, Fumin Zhang. Literature research on related areas of the project topic has been performed. Team members have met and gained basic experience of controlling the blimp via a joystick. Currently, each team member is picking up related software including TensorFlow, ROS, and OpenCV, if he or she does not have related experience.

9. Leadership Roles

According to the specialty of each team member, different leadership roles are assigned as in Table 6.

Table 6. Leadership Roles

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

10. References

[1] M. Maimone, Y. Cheng, and L. Matthies, “Two years of visual odometry on the mars exploration rovers,” J. Field Robotics, vol. 24, pp. 169–186, 03 2007.

[2] L. Z. Wei, “Lessons for deep-tech startups from skydio, the autonomous drone company,” Feb 2018, accessed: 2019-09-29. [Online]. Available: https://medium.com/zhanwei/lessons-for-deep-techstartups-from-skydio-the-autonomous-drone-company-61fa890ea990

[3] Stever, “Dji matrice 100 & dji guidance announced,” Jun 2015. [Online]. Available: https://www.heliguy.com/blog/2015/06/09/dji-matrice-100-dji-guidance-announced/

[4] G. Zhou, Lu Fang, Ketan Tang, Honghui Zhang, K. Wang, and K. Yang, “Guidance: A visual sensing platform for robotic applications,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), June 2015, pp. 9–14.

[5] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017, pp. 6612–6619. [Online]. Available: https://github.com/tinghuiz/SfMLearner

[6] S. Sumikura, M. Shibuya, and K. Sakurada, “OpenVSLAM: A Versatile Visual SLAM Framework,” in Proceedings of the 27th ACM International Conference on Multimedia, ser. MM ’19. New York, NY, USA: ACM, 2019. [Online]. Available: https://doi.org/10.1145/3343031.3350539

[7] D. Scaramuzza and F. Fraundorfer, “Visual odometry: Tutorial,”IEEERobotics & Automation Magazine, vol. 18, no. 4, pp. 80–92, 2011.

[8] F. Fraundorfer and D. Scaramuzza, “Visual odometry: Part II: Match-ing, robustness, optimization, and applications,”IEEE Robotics &Automation Magazine, vol. 19, no. 2, pp. 78–90, 2012.

[9] S. Wang, R. Clark, H. Wen, and N. Trigoni, “Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks,” in 2017 IEEE International Conference on Robotics and Automation (ICRA), May 2017, pp. 2043–2050.

[10] I2C (Inter-Integrated Circuit) Framework, 2018 [Online]. Available: http://www.qnx.com/developers/docs/6.6.0.update/index.html#com.qnx.doc.neutrino.technotes/topic/i2c_framework.html

[11] USB – Universal Serial Bus 3.0 and 2.0 Specifications [Online]. Available: https://www.intel.com/content/www/us/en/products/docs/io/universal-serial-bus/universal-serial-bus-specifications.html

Appendix A

Senior Design Project Skills Matrix

ECE4011 rev.1 10/19/2019

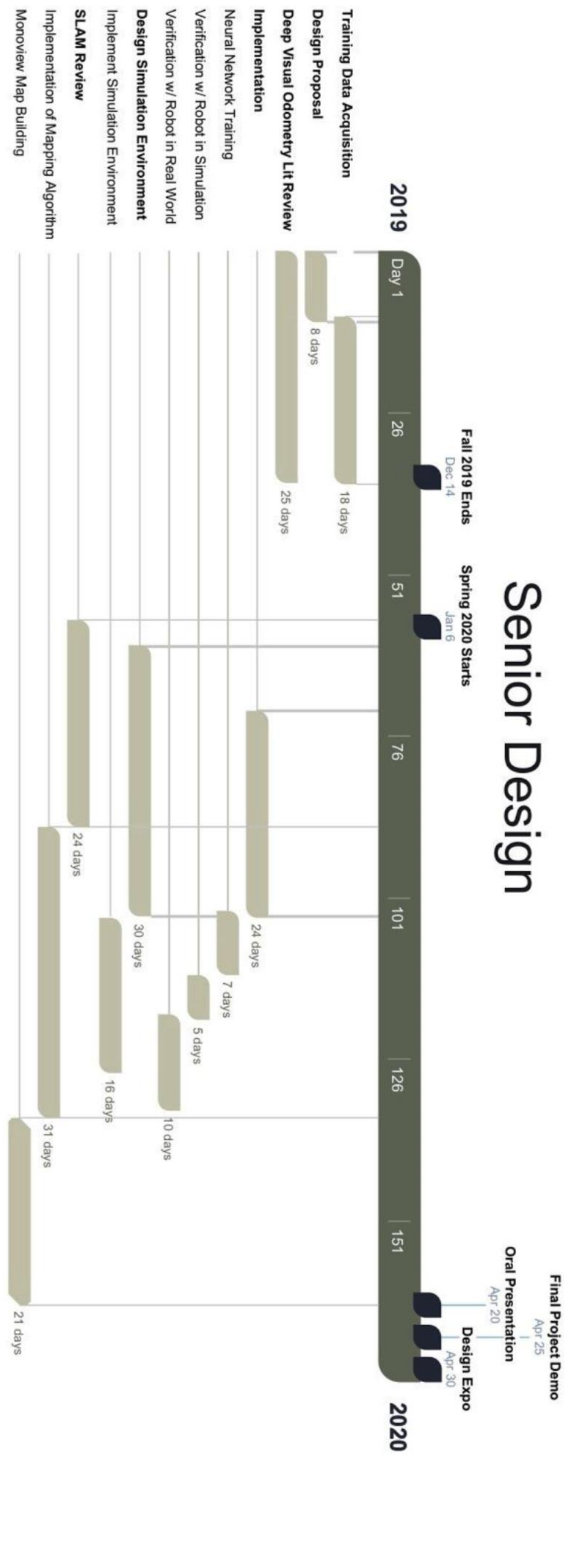
Skill Levels Normalized to μ=0 s=1 for each member with avg team skill added

Team Member	Machine Learning Implementation	Computer Vision System Building	Real-Time Software Coding	PCB or Mechanical Design	Graphical User Interface Coding	Web Application Development	Technical Marketing	Project Management	Team Leadership
Yifan Shen	3.3	1.4	3.3	-0.7	2.5	2.0	0.9	1.1	2.0
Fanzhe Lyu	3.3	4.1	4.6	0.6	2.5	0.6	2.3	2.5	0.6
Yilun Xie	4.6	2.8	3.3	0.6	2.5	0.6	0.9	2.5	0.6
Ruoyang Xu	4.6	2.8	4.6	3.4	1.1	0.6	2.3	2.5	0.6
Average Skill Level	4.0	2.8	4.0	1.0	2.2	1.0	1.6	2.2	1.0
Specialist Rating	0.8	1.1	0.8	1.7	0.7	0.7	0.8	0.7	0.7
Analysis of Team Skills	Strong	Strong	Strong with Multiple Specialists	Average with Specialists	Average	Weak	Weak	Weak	Weak

Avg(StDev)= 0.89

Suggestions for Team Assignments	
Machine Learning Implementation	Yifan Shen
Computer Vision System Building	Fanzhe Lyu
Real-Time Software Coding	Yifan Shen
PCB or Mechanical	Ruoyang Xu
Graphical User Interface Coding	Yilun Xie
Web Application Development	Yifan Shen
Technical Marketing	Fanzhe Lyu
Project Management	Yilun Xie
Team Leadership	Ruoyang Xu

Appendix B


Senior Design
2019
2020
Training Data Acquisition
Design Proposal
Deep Visual Odometry Lit Review
Implementation
Neural Network Training
Verification w/ Robot in Simulation
Verification w/ Robot in Real World
Design Simulation Environment
Implement Simulation Environment
SLAM Review
Implementation of Mapping Algorithm
Monoview Map Building
Day 1
26
51
76
101
126
151
Fall 2019 Ends
Dec 14
Spring 2020 Starts
Jan 6
Oral Presentation
Apr 20
Final Project Demo
Apr 25
Design Expo
Apr 30
8 days
18 days
25 days
24 days
7 days
5 days
10 days
30 days
16 days
24 days
31 days
21 days