

McMASTER UNIVERSITY

DEVELOPMENT PROCESS & IMPLEMENTATION

4G06 CAPSTONE DESIGN PROJECT

3D Scanning Quadcopter

A Large Scale 3D Scanner

Paul Correia
Mechatronics Eng
1132370

Nicolas Lelievre
Software Eng
1203446

Bennett Mackenzie
Software Eng
1211985

Tigran Martikian
Software Eng
1213170

Balraj Shah
Software Eng
1207997

Mykola Somov
Software Eng
1141160

October 20, 2015

Contents

1	Project Proposal Revision	2
1.1	Project Description	2
1.2	Required Skills & Engineering Knowledge	2
1.3	Group Composition	2
1.4	Clarifications	3
2	Roles & Responsibilities	4
2.1	Paul Correia	4
2.2	Nicolas Lelievre	4
2.3	Bennett Mackenzie	4
2.4	Tigran Martikian	4
2.5	Balraj Shah	5
2.6	Mykola Somov	5
3	Process Workflow	6
3.1	Steps	6
3.2	Inputs & Outputs	7
3.3	Acceptance Criteria	7
4	Process Tools	8
5	Version Control	9
5.1	Using Version Control	9
5.2	Changes in Development Artefacts	9
5.2.1	Changes and Bugs	9
5.2.2	Deposition	10
5.2.3	Bug Deposition	10
5.2.4	Change Deposition	11
5.3	Accessing Version Control	11

1 Project Proposal Revision

1.1 Project Description

We propose for our capstone project, a quadcopter that can scan an object (tentatively around the order of one to two meters, possibly scaled higher) and produce a 3D model that can be scaled down and 3D printed.

Our initial impressions are to have the quadcopter scan the object (using either image processing, lidar detection or a combination thereof) by circling said object to be scanned and convert the captured data to a 3D model file for examination and potentially 3D printed at a scaled down size.

As far as our research has revealed, no large scale automated three dimensional scanning peripherals currently exist, giving our project both meaning in versatile and real life application as well as focus and exposure within an emerging industry.

1.2 Required Skills & Engineering Knowledge

Our group will necessitate knowledge in a variety of fields including basic quadcopter operation, real-time flight automation, real-world to model mapping conversion and model scaling/3D printing.

Luckily, both quadcopters and 3D printers have reached a certain level of popularity in numerous fields resulting in high levels of documentation, scalability and customisation with minimal resources required.

The challenge will rest with automating the quadcopter's control and operations with regard to properly scanning a large-scale physical object with reliable and sufficient accuracy to then be converted into digital 3D model formats suitable for printing.

These necessary skills and knowledge base, however, do not surpass our abilities as engineers.

1.3 Group Composition

We believe our group composition is very capable of this task.

The level of mechanical expertise necessary to complete this project is well within our abilities, given that quadcopters are well-documented in terms of design and customisability. Also note that we have a mechatronics student to aid with both electrical and physical design.

Software design will be the bulk of the challenge of this project. Specifically, generating the 3D model from the sensor(s) as well as the control software utilised to fly and automate the quadcopter's operations. The software itself will be where most of the complexity lies, which is fitting due to the fact that the majority of our group consists of software engineers.

Planning ahead, our proof of concept due in December could be a functioning quadcopter and/or proof that valuable information is gatherable and accurate enough for use in generated 3D models.

1.4 Clarifications

A few clarifications regarding our project. The project will consist of a drone with a camera on board. It will store a 360 degree set of images of the object being scanned. In the event that the object is taller than the camera's field of vision (with respect to the object's height) the drone will move into position for multiple orbits around the object until it reaches the top of the scan. This information will then be used in order to render a 3D model of the object which will then be available for 3D printing.

A key aspect of this project is the removal of the human element. The drone will be autonomous in the sense that once it begins scanning there will be no further need for human interaction. It will also allow humans to scan objects too large or irregular in shape for manual scanning.

2 Roles & Responsibilities

Below are the projected roles and responsibilities for all members of the group. Please note that these are not exclusive and members are expected to contribute to a variety (if not all) aspects of the final product.

2.1 Paul Correia

Paul Correia will be working on the Quadcopter team, specifically mechanical design and specifications as well as electrical circuit design and power systems. He will be collaborating with Ben Mackenzie and Nicolas Lelievre on the Quadcopter's overall functionality.

His experience in robotics design and testing will allow him to design reliable electromechanical systems with appropriate safety tolerances. Paul will also contribute to impartial testing and verification of all non-embedded software systems, specifically the image processing systems.

2.2 Nicolas Lelievre

Nicolas Lelievre will be working with Paul Correia and Ben Mackenzie in concentrating on the quadcopter's overall functionalities. More specifically, he will help design, build and test the quadcopter from basic parts as well as design the quadcopter's launchpad.

His experience in software engineering and control systems will allow him to assist in the flight controller's design and testing as well as tend to the above mentioned tasks. Initial testing of the quadcopter will be conducted by the quadcopter team, however more elaborate tests will involve members of the 3D scanning team in order to ensure proper separation of tasks.

2.3 Bennett Mackenzie

Ben Mackenzie will be working on the Quadcopter team, specifically Quadcopter Design and Control Systems Design for the Quadcopter. He will be collaborating with Paul Correia and Nicolas Lelievre on this aspect of the project. He has experience working with electrical design on robots and working with software, making him a good fit for the job.

Preliminary testing and analysis of Ben's work may be done by the Quadcopter team, followed by a walkthrough with the Sensor team to ensure impartial review of any design.

2.4 Tigran Martikian

Tigran Martikian will be working on the development of the 3D model. As part of the sensor and image processing team, alongside Balraj Shah and Mykola Somov, we will be working together to take 2D images and convert them into

a 3D model. Tigran, will also be assisting with data transmission between the sensor and quadcopter

He will also assist in testing the quadcopter and sensors. Further testing and verification will be done by all members of the group.

2.5 Balraj Shah

Balraj Shah will be working on the development of the sensor portion of this project with Tigran Martikian and Mykola Somov. He has experience with software development which will make him an asset in creating a method/program for extracting a 3D model from a set of 2D images. Furthermore, he will assist in the transmission of data between the sensor and any storing device.

The preliminary testing and verification will be done by all members working on the sensor portion. Further verification of functionality and any overlooked aspect in regards to the sensor will be done by the Quadcopter team. Testing and verification of data transmission will be done by both groups collaboratively.

2.6 Mykola Somov

Mykola Somov will be working with Tigran Martikian and Balraj Shah for the explicit purpose of designing the software and mechanism of the sensor functionality of the quadcopter. He will prioritize his work in modifying existing software designs and algorithms to meet the sensor needs of the quadcopter's overall functionality. Mykola will also contribute to preliminary functional testing as well as code revision.

Mykola's education in software engineering as well as industry experience in functional testing of code will allow him to perform these tasks with a high degree of quality and professionalism.

3 Process Workflow

The workflow process described in the following sections is subject to change throughout the project and should not be taken as a final structure.

3.1 Steps

Due to the fact that this project comprises of several distinctly different parts, most of the development will be achieved concurrently. This being said, the bulk of the endeavour may be divided into two main sections: a functioning quadcopter and a functioning large-scale three-dimensional scanner. The two halves will be integrated within one another once they have reached acceptable independent functionality.

1. **Necessary Parts** Processes necessary to determine what hardware and software will be required for the quadcopter and scanner.
 - (a) Design feasible quadcopter
 - (b) Design feasible 3D scanner
 - (c) Cost analysis
 - (d) Acquire (purchase or make) all necessary parts
2. **Quadcopter** Designing a fully functional and automated quadcopter.
 - (a) Assemble basic quadcopter capable of sustaining all component weight
 - (b) Add quadcopter flight controller
 - (c) Manipulate flight controller to allow automated flight trajectories
 - (d) Design launchpad for distance and position locating as well as homing
3. **Three-Dimensional Scanner** This step applies to the 3D scanner and all applications necessary to render it functional.
 - (a) Assemble basic 3D scanner capable of scanning objects
 - (b) Ensure 3D scanner functions in continuous motion (ie. during flight)
 - (c) 3D scanner returns usable and accurate data
 - (d) Convert data into 3D model
4. **Integration** This final step includes the integration of the functioning quadcopter and 3D scanner.
 - (a) Secure 3D scanner onboard quadcopter
 - (b) Retrieve usable and accurate data from flight
 - (c) Convert 3D scans to 3D models
 - (d) Print model using 3D printer

Note that the above steps are a general high-level overview and do not take into account any necessary revisions or extensive testing made along the way. Also note that steps 2 and 3 (quadcopter and three-dimensional scanner, respectively) will be designed concurrently within the group.

3.2 Inputs & Outputs

The following is a high-level overview of the inputs and outputs required for a functioning system.

input	<i>Proximity Sensor Distance</i>	Necessary to keep a safe distance from scanning object.
input	<i>Launchpad Distance</i>	Necessary to detect when the device has reached the initial launch position (full circumference scanned).
input	<i>Camera Pictures</i>	Necessary to analyse and create three-dimensional model.
output	<i>3D Model</i>	Result of all above inputs, the model should be an accurate digital representation of the real-world object.

3.3 Acceptance Criteria

The acceptance criteria for the system relies entirely on the final output of the system and, as such, will be evaluated based on the resulting 3D model generated from all inputs.

Hence, an acceptable 3D model will be one that can be easily recognised by users familiar with the scanned object (either with, or without any colour mapping rendered on the surface of the model).

4 Process Tools

The below process tool will be used in the design and implementation of the large-scale 3D scanning quadcopter.

Arduino 1.6	For embedded controls applications, we will be using Arduino due to its simplicity and ease of use.
Autodesk Inventor 2014	For the hardware design of the quadcopter, we will be using Autodesk Inventor. We chose this over other 3D CAD software due to our past experience using Inventor.
CadSoft EAGLE 7.4	If we need to design any custom PCBs, we will use CadSoft's EAGLE due to large library of standard components and ease of exporting for fabrication. The tolerances to which the PCB will be designed to, i.e, trace width, minimum drill size, etc., will depend on the manufacturer's design rules.
Eclipse 4.5	We will be using Eclipse as our IDE for non-embedded programming. Eclipse was chosen due to our group's familiarity with the Java programming language and Eclipse itself.
git 2.0	We will be using git for version control for reasons discussed in section 5.1
OpenCV 2.4	We will be using OpenCV for any image processing. We decided to use version 2.4.10 as opposed to the latest stable release 3.0.0 since this new release doesn't provide us with any vital new features. 2.4.10 is a mature release with very few open bugs and as such is well suited to our application.

5 Version Control

5.1 Using Version Control

As a group, we were deciding whether we should use SVN or Git to manage our version control. In the end we chose to use Git because we are familiar with github and its socially interactive coding experience.

One of the reasons we are using GitLab is it allows us to collaborate on a single codebase effectively. GitLab also allows us to resolve merge conflicts easily. It gives us the opportunity to work in our own branches and use a pull request system to get feedback on code before it is merged. As a group, we can discuss, comment, and improve each other's code in these pull requests. Git allows us to have a complete history of all our code and every version of each commit. Lastly, if something goes wrong in the future of our project, we can simply revert back to a working version from a previous date.

5.2 Changes in Development Artefacts

The following section details the means by which change requests and bug reports are documented and serviced. All bug reports, and change requests will be submitted on the project's gitlab page as an issue. The following section is not liable to change under any but the most extreme of circumstances. In such a case, the nature of the revision and its justification will be specified in the project's gitlab page as an issue with the 'MASTER CHANGE ORDER' label.

5.2.1 Changes and Bugs

All change requests and bug reports must be submitted as an issue with the most representative of one of the following identifications included as a label:

- E1 A system critical bug which prevents the quadcopter from achieving controllable flight
- E2 A critical bug which prevents system requirements from being met
- E3 A bug which does not directly violate system requirements but induces strange system behaviour that may negatively impact user experience
- E4 A non-critical bug which causes unexpected system behaviour but is not likely to negatively impact user experience or violate system requirements
- E5 A suspicion of the existence of a bug
- C1 A change request to the system
- C2 A concern with the system without a change request specified

All new change requests and bug reports must include on what build and when the issue was first found. Bug reports must include instructions on how the bug was triggered as well as a copy of the error log if available. A bug report may also have its identification change with testing. In such a case the change, reason for change and time of change will be specified in a comment to the report. Change requests must include justification as to why a change should be applied.

5.2.2 Deposition

All change requests and bug reports must be reviewed by at least one project member whom did not originally report the issue, within 48 hours of the issue being posted. A reviewer must begin the process of deposition of E3 and higher level bug reports within 72 hours of reviewing the report. All other issues raised must begin the process within two weeks of review. The process of deposition is outlined below:

5.2.3 Bug Deposition

1. After a bug report has been reviewed, a reviewer must comment on the issue to denote that they have begun working on the issue
2. The reviewer must replicate the bug at least once and then add the 'confirmed' label to the issue
 - (a) If the bug cannot be replicated, the reviewer must add the 'unconfirmed' label to the issue instead and temporarily cease the deposition process
 - (b) If a reviewer is later able to replicate the bug, the 'unconfirmed' label must be replaced with a 'confirmed' label and the deposition process continues
3. Once a bug is confirmed the group must decide on a course of action based on the bug's severity
 - (a) E2 and higher level bugs should have at least three group members prioritizing releasing a fix
 - (b) E3 bugs should have at least one group member prioritizing releasing a fix
 - (c) Bugs of lower severity than E3 can be ignored until such a time that their severity increases higher than E4
4. When fix is ready, it is applied to the build, and a group member must comment on the bug indicating which build was applied
5. A group member reviews the new build to determine if they could still replicate the bug

- (a) If the bug cannot be replicated the reviewer must comment on the issue indicating what build fixed the bug and close the issue
 - (b) If the bug can be replicated the reviewer must comment that the current build did not fix the bug, and the deposition process returns to step (2) until the issue can be closed
6. If at a later time a group member is able to confirm that the same bug is still prevalent, the group member must reopen the issue, note what build the bug was rediscovered on, and under what conditions it occurred. The deposition process restarts in this case

5.2.4 Change Deposition

1. After a change request has been reviewed by at least four group members, all reviewers and the original poster should discuss the proposed changes
2. During discussion amendments can be made to the proposal
 - (a) Once at least four group members are satisfied to the proposal, a comment must added indicating any amendments made to the proposal, the 'okay' label must be added, and at least one group member should be assigned to make the changes
 - (b) Alternatively if five group members opt to veto the proposal, no changes are applied, the 'not applied' label is added to the proposal, and the proposal must be closed
3. Once the changes are ready, they are applied to the build, and a comment must be added to proposal indicating what the new build this is
4. The group must then evaluate the new build to see if the new build satisfies the amended proposal
 - (a) If the new build is satisfactory, the 'okay' label must be replaced with the 'applied' label, and the issue must be closed
 - (b) If the new build is unsatisfactory, the deposition process returns to step (1)

5.3 Accessing Version Control

We are using GitLab for our version control system as previously mentioned.

The repository for our project is available at:

https://gitlab.cas.mcmaster.ca/mackeb/Group_13_Capstone.git

Please contact mackeb@mcmaster.ca if any person needs to be given access to the repository.