**ENGI 3051 – Software Engineering Design II**

**Fall 2015**

**Stereoscopic 3D Reconstruction System**

Software Specifications Document

Contributing Members:  
Slim Babay, Antoine Stavro

Date Submitted: 10/16/2015

# Approval

This document has been read and approved by the following team members responsible for its implementation:

Developers

|  |  |  |
| --- | --- | --- |
| **Print Name** | **Signature** | **Comments** |
| Slim Babay |  |  |

|  |  |  |
| --- | --- | --- |
| **Print Name** | **Signature** | **Comments** |
| Antoine Stavro |  |  |

Client

|  |  |  |
| --- | --- | --- |
| **Print Name** | **Signature** | **Comments** |
| Greg Toombs |  |  |

# Revisions

The project has been renamed from Stereo Camera 3D Reconstruction to Stereoscopic 3D Reconstruction System to clarify the interaction between the software and hardware. We agreed that the name would better reflect our methodology on using 2 separate cameras as opposed to a fully integrated stereo camera.

# Table of Contents

[Approval 2](#_Toc432794692)

[Revisions 3](#_Toc432794693)

[Table of Contents 4](#_Toc432794694)

[Introduction 5](#_Toc432794695)

[Project Overview 5](#_Toc432794696)

[Glossary 5](#_Toc432794697)

[Product Specifications 7](#_Toc432794698)

[UML Functional Model 8](#_Toc432794699)

[Capture from Cameras (Use case) 8](#_Toc432794700)

[Import 3D model 9](#_Toc432794701)

[Import nonexistent file (Extends: Import 3D model) 9](#_Toc432794702)

[Export 3D model 9](#_Toc432794703)

[Rotate 3D Plane 10](#_Toc432794704)

[UML Class Diagram 10](#_Toc432794705)

[Graphical User Interface Draft 11](#_Toc432794706)

[Software Project Management Plan 12](#_Toc432794707)

[Software Process Model 12](#_Toc432794708)

[Deliverables and Milestones 12](#_Toc432794709)

[Roles and Responsibility 13](#_Toc432794710)

[Leadership 13](#_Toc432794711)

[Development Effort: Measured in man-months Intermediate COCOMO 13](#_Toc432794712)

[Description of Risks and Risk Mitigation 15](#_Toc432794713)

[Software scheduling flaws 15](#_Toc432794714)

[Poor Productivity 15](#_Toc432794715)

[List of Figures and diagrams 16](#_Toc432794716)

# Introduction

This software specification document serves to provide a detailed overview of the software solution being developed for the client Greg Toombs. A 3D capturing and displaying application is to be developed using Java and object oriented methodologies. The software can be divided into 2 separate components: a backend that handles the 3 dimensional reconstruction from images acquired by at 2 separate cameras and a frontend web application that provides a graphical user interface and displays the rendering of the 3 dimensional object produced by the backend. Additional detailed step-by-step descriptions of each usecase from the previous requirements document are also provided for further explanation.

## Project Overview

The Stereoscopic 3D Reconstruction System will be deployed on a linux server connected to at least 2 USB cameras. It will allow users to connect through a web application to capture separate images from each camera and reconstruct a 3D model by utilizing the stereoscopic reconstruction functions within the open source computer vision library, OpenCV. The 3D model produced by the system will be rendered using WebGL, a web graphics library, to allow for seamless rotation and 3D manipulation of the model. The user will also be able to import and export 3D models.

## Glossary

Table - Glossary of terms pertaining to the domain of the project

|  |  |
| --- | --- |
| Term | Description |
| Focal Length | The optical distance where light rays converge to form a sharp image of an object to the camera sensor at the focal plane in the camera. The focal length is one of the parameters determining the perspective of the image produced by the camera. |
| Perspective | The visual effect or difference in the apparent sizes of objects that are at different distances from the camera. Given 2 images taken at the same location, changing the focal length of lens will also change the angle of view and magnification of the image. |
| Depth of Field | The distance that appears to be in focus in front of and behind the focal point of the lens. This is determined by the aperture, lens focal length, and the distance to the subject. |
| Calibration | Function served to form the scene view by projecting 3D points into the image plane using a perspective transformation of a known model. To simplify the process for the end user, most of this will be done during development with additional adjustments performed based on inter-image correspondence matching  Matrices of intrinsic and extrinsic parameters are produced to offset and normalize factors that would account for errors in stereoscopy. |
| Intrinsic parameters | Internal variables dependent of the source of the image. These include varying factors such a focal length, lens distortion and the principal point offset between different cameras. It creates a mapping between the pixel and camera coordinates in the frame. |
| Extrinsic parameters | External variables dependent on the position and rotation of camera. These include difference perspectives of the scene of the difference cameras and are required for stereoscopy. |
| Object rendering | It is the process or function of generating an image based on geometry of the 3 dimensional model and the viewpoint or perspective it is being viewed from. |
| Point Cloud | A point cloud is a set of data points (X, Y and Z) of a 3D coordinate system used to form the model. Point represent the external surface of the model and are derived from corresponding matches from images and their calculated location in space. |
| Aperture | Aperture is a hole or an opening through which plays a role in determining how much light reaches the image plane. In the context of this project, it is equivalent to the size of the image sensor of the specified camera. |
| Aspect Ratio | Description of the the proportional relationship between the width of an image and its height. In the context of this project it is a function of the height divided by its width. |
| Epipolar lines | A line vector that corresponds points in one image to another. |

# Product Specifications



Figure - Illustrated system overview of the Stereoscopic 3D Reconstruction System.

## UML Functional Model

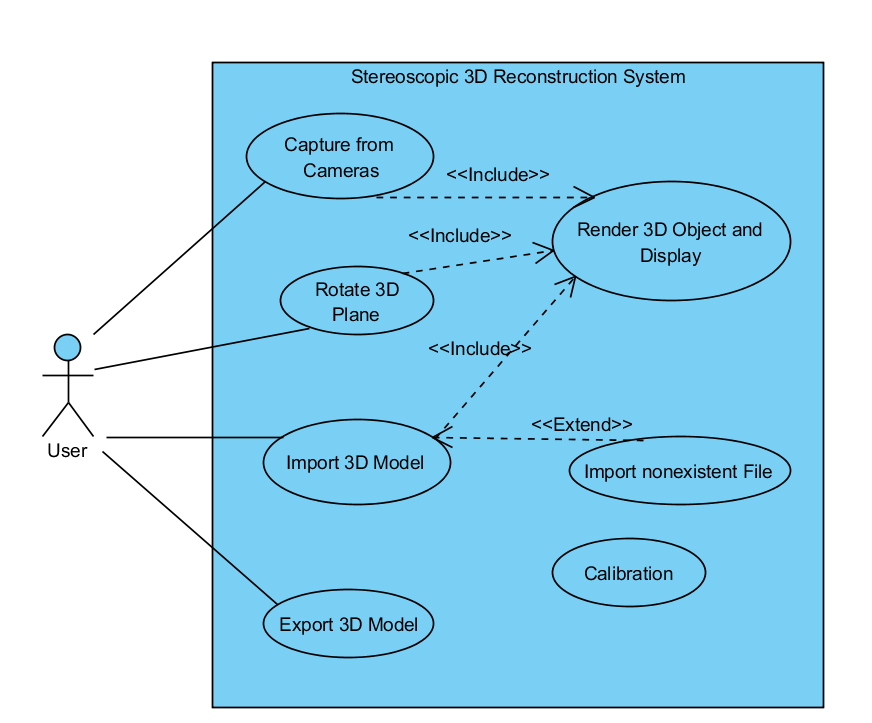


Figure –Use case diagram of the Stereoscopic 3D Reconstruction System

### Capture from Cameras (Use case)

|  |
| --- |
| **Brief Description**  The c*apture from cameras* use case will allow the user to capture the 3D object from the camera. Images from at least 2 cameras are taken and used to reconstruct the 3D model. The client is then sent all the images to be displayed and rendered, respectively, in their browser. |
| **Step-by-Step Description**   1. Setting up object    1. The user will setup the camera apparatus and position it in such a way that the cameras are facing the desired object.    2. With the server up and running, the client will visit the web application in their browser. 2. Once the cameras are in their desired spot, the user can click on the Capture button which will instruct the server to save the images from the cameras and begin the reconstruction process. 3. Once the server has reconstructed the 3D model, it will send the client the images from each camera, the 3D model to display to the user, and a URL to where the exported model can be downloaded. |

### Import 3D model

|  |
| --- |
| **Brief Description**  The *import 3D model* use case will allow the user to use the web application to open and view a 3D model from a specified file on the client computer. |
| 1. The client will visit the web application in their browser.    1. To load a 3D model, the user will click on the “Import” button.    2. An open file dialog window will pop up to assist the user in finding the file that they would like to view.    3. Once the file the has been located, the user will click “open” as part of the html file api. 2. The browser will use the file path to open the model for rending in the browser. |

### Import nonexistent file (Extends: Import 3D model)

|  |
| --- |
| **Brief Description**  The *import 3D model* use case occurs if the user specifies a file that does not exist or if a problem occurs in accessing the file. |
| 1. The client will visit the web application in their browser.    1. To load a 3D model, the user will click on the “Import” button.    2. An open file dialog window will pop up to assist the user in finding the file that they would like to view.    3. Once the file the has been located, the user will click “open” as part of the html file api. 2. The file api will throw a javascript exception if there is a problem. Our web application will catch the exception and will notify the user that there was a problem attempting to open their file. |

### Export 3D model

|  |
| --- |
| Brief Description  The *export 3d model* use case will allow the user to save the 3D object on their computer. Since the reconstruction software will be on a server, the model will be automatically saved in the backend and exported to a cache that can be cleared out periodically. A URL pointing to its location would be sent after they instruct the system in capturing the 3d model and which can be access by clicking on the “export” button. This will allow the user to seamlessly allow their browser to download the file. |
| 1. After having captured a model from the cameras, the “export” button will become enabled once the model has been shown. 2. The user can click the “export” button which will point their browser to the 3D model file URL on the server causing the clients browser to begin downloading it. |

### Rotate 3D Plane

|  |
| --- |
| Brief Description  The *rotate 3d plane* use case in a function that will allow the user to rotate the 3D model. By using the mouse, the user can rotate and observe the model from all angles. |
| 1. Once the user has a 3D model in the viewing area, the will be able to use their mouse to interact with it. 2. To rotate Model:    1. The user will click and hold the mouse button down, and drag the mouse away from its original position.    2. As the mouse drags, the 3D model will rotate in the direction of the mouse in relation to its movement |

## UML Class Diagram

The following diagram is a representation of the classes used by the stereoscopic 3D reconstruction system. The software will be designed to modularly accommodate for the capability that the opencv library has in supporting multiple cameras. This will allow further development and addition of new features and updates if continued support is desired after the software has been delivered to the client.

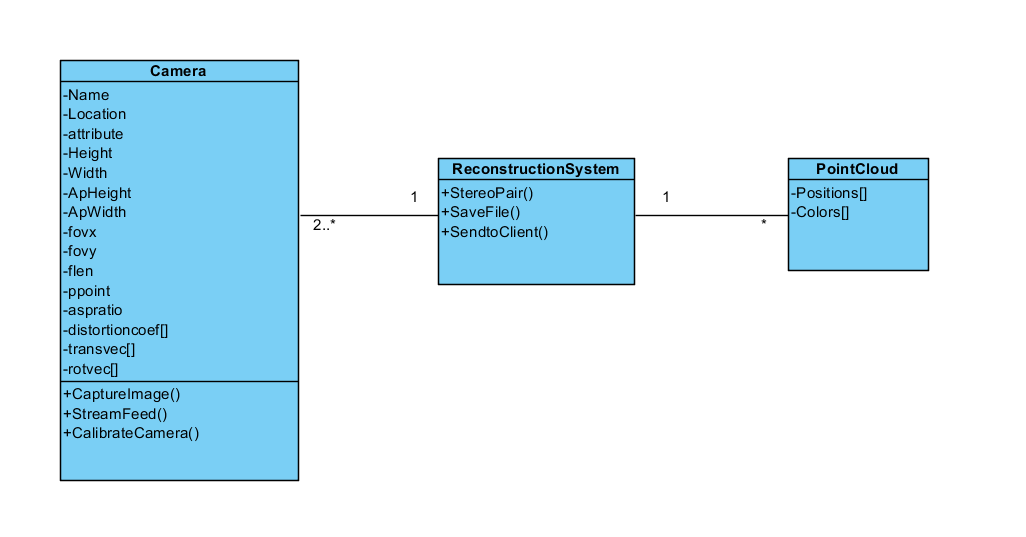


Figure - UML Class Diagram of the Stereoscopic 3D Reconstruction System

## Graphical User Interface Draft

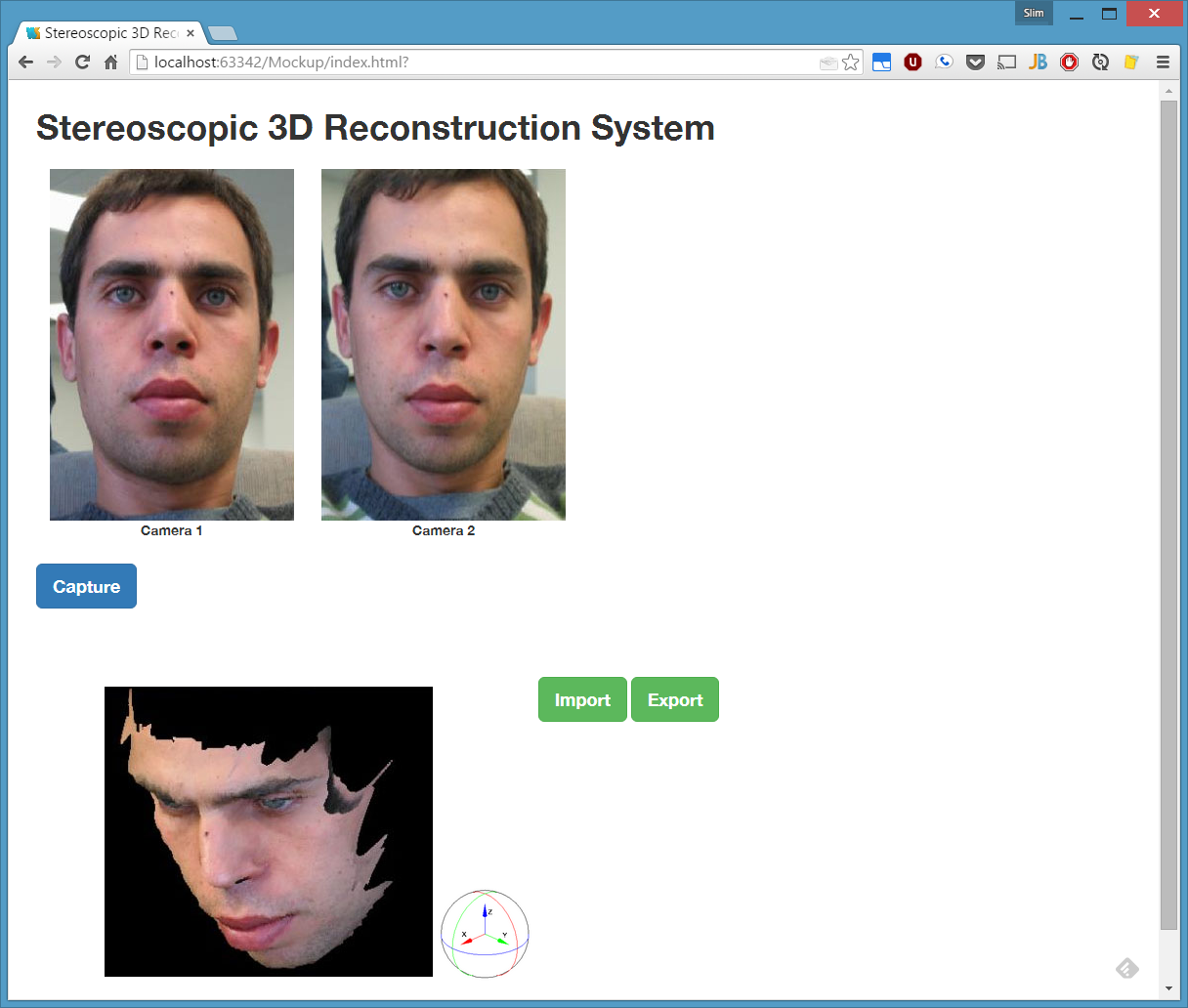


Figure - Mock up GUI of the Web application

# Software Project Management Plan

## Software Process Model

The software development process model that will be used to develop the stereoscopic 3D reconstruction system will be the agile methodology with an object oriented design and iterative and incremental approach. The agile method was chosen because of it’s emphasis on quickly developing a minimum viable product and constant product evolution.

Making the most of an agile software process model will require the following

* Regular team communication in the form of short SCRUM meetings; short meetings where team members discuss current progress and any roadblocks that have come up.
* Pair programming that will allow for higher quality code and will promote the passing of knowledge and experience between team members.
* Self organizing and flexible teams. Team members may contribute to other parts of the development process that may be outside their initial responsibilities.
* Early development of a working product that will be improved in later product versions.

## Deliverables and Milestones

This is a compilation of all the important requirements for the successful completion of the project:

Table - Project management task list

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Start Date | Duration (Days) | End Date |
| Software Requirement Document | 21-Sep | 11 | 2-Oct |
| Software Specification Document | 3-Oct | 13 | 16-Oct |
| Download/install required software | 12-Oct | 4 | 16-Oct |
| Acquire and test 2 camera in Ubuntu | 16-Oct | 1 | 17-Oct |
| Calibration and implementation of OpenCV | 18-Oct | 7 | 25-Oct |
| Test Calibration with objects | 26-Oct | 2 | 28-Oct |
| Software Design Document | 17-Oct | 6 | 23-Oct |
| Design GUI | 24-Oct | 11 | 4-Nov |
| Implement Mouse Rotation in GUI | 31-Oct | 10 | 10-Nov |
| Software Testing Document | 24-Oct | 20 | 13-Nov |
| Unit testing | 14-Nov | 18 | 2-Dec |
| Integration testing | 14-Nov | 18 | 2-Dec |
| Implementation and Final Demo | 2-Dec | 0 | 2-Dec |

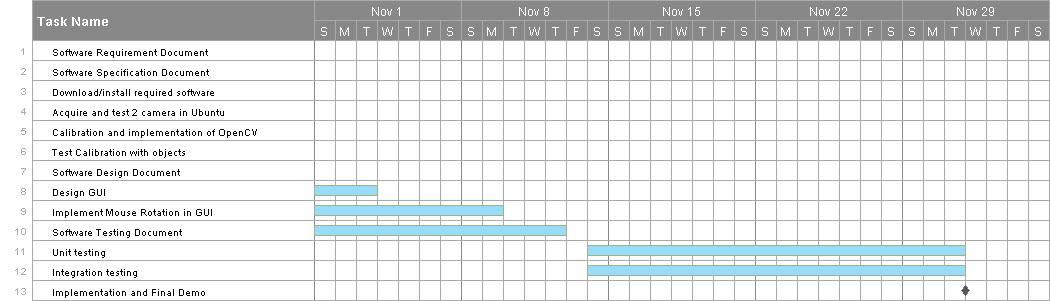
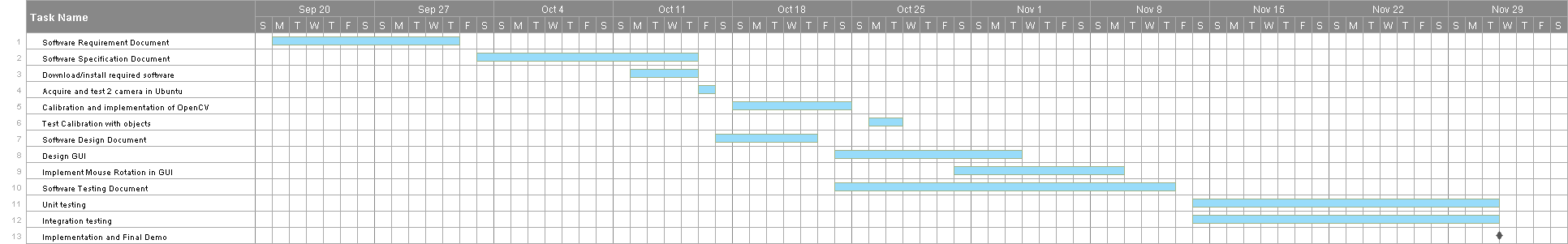


Figure - Gantt chart of the project milestones and subtasks

## Roles and Responsibility

Although all members of the team will implement the entire software, the software can be split into 2 subsystems capable of running semi-independent of each other. Slim Babay will lead in developing the web application component of the system while Antoine Stavro will head integration of the reconstruction API.

## Leadership

Project leader role will ensure that the development process is going smoothly. They are responsible for keeping on top of scheduling, task management, work tickets, and hosting regular SCRUM meetings. It will be a role similar to a project manager but will also be involved in the development and testing process along the way.

The project leader for the first half of the project will be Antoine Stavro. Half way though the project (Nov 10) the project leader position will be taken over by Slim Babay. At the end of each person’s turn being project leader they will document what has been done and is to be done for the rest of the team members and the new project leader.

## Development Effort: Measured in man-months Intermediate COCOMO

Nominal Effort = a x ( KDSI )ᵇ Person-Months

For our project we’ll be using the Organic Product model because our software will be smaller than 5000 LOC. The program will contain roughly 1.5 KDSI, because the program we are making uses a lot of libraries functions. Also, there will be a lot of integration involved with our program.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | a | b | KDSI | Total |  |
| Nominal Effort | 3.2 | 1.05 | 1.5 | 5 | person-months |
| Nominal Effort | 5 x 1.22 | | | 6 | person-months |

|  |
| --- |
| organic |
| a = 3.2 |
| b = 1.05 |

*The intermediate COCOMO calculation for this software development comes out to be 6 man-months. This seems to be a fair estimate since this project is meant to be finished by 2 students working for a total of approximately 3 to 4 months.*

|  |  |  |
| --- | --- | --- |
| Cost Drivers | Rating | |
|  |  |  |
| **Product Attributes** |  |  |
| Required software reliability | 1.4 | High |
| Database size | 0 | Very low |
| Product complexity | 1.15 | High |
|  |  |  |
| **Computer Attributes** |  |  |
| *Execution time constraint* | 1.11 | High |
| *Main storage constraint* | 0 | Low |
| *Virtual Machine volatility* | 0 | Very low |
| *Computer Turnaround time* | 1.07 | High |
|  |  |  |
| **Personal Attributes** |  |  |
| *Analyst capabilities* | 1 | Nominal |
| *Applications experience* | 0.91 | High |
| *Programmer capability* | 0.86 | High |
| *Virtual Machine experience* | 0.95 | High |
|  |  |  |
| **Project Attributes** |  |  |
| *Use of modern programming practices* | 0.91 | High |
| *use of software tools* | 0.91 | High |
| *Required development schedule* | 1.04 | High |
|  |  |  |
| **Total Sum** | **1.22** | |

Figure - Intermediate COCOMO metric

## Description of Risks and Risk Mitigation

The following section outlines some of the possible risks involved in the development of the Resource Management System and ways that each of them can be mitigated.

### Software scheduling flaws

Team members with individual schedules can make determining the estimated development time difficult and could lead to inaccurate estimates.

**Solution:** Scheduling will be done by all members of the team together as opposed to being laid out by a single person which should lead to better time estimations.

### Poor Productivity

Long term goals can lead to can lead to a lose of the sense of urgency in the team and work can pile up.

**Solution:** Working in sprints with exact goals planned to achieve will help keep an accurate schedule of production. Planning the length of sprints and having them at regular intervals will ensure staying on schedule.

# List of Figures and diagrams

[Figure 1 - Illustrated system overview of the Stereoscopic 3D Reconstruction System. 7](#_Toc432794717)

[Figure 2 –Use case diagram of the Stereoscopic 3D Reconstruction System 8](#_Toc432794718)

[Figure 3 - UML Class Diagram of the Stereoscopic 3D Reconstruction System 10](#_Toc432794719)

[Figure 4 - Mock up GUI of the Web application 11](#_Toc432794720)

[Figure 5 - Intermediate COCOMO metric 14](#_Toc432794721)

(This page left intentionally blank)