University of Cape Town

Undergraduate Thesis

Determining Scale within Unstructured Point Clouds

Author:
Jayren Kadamen

Supervisor:
Dr. George Sithole

A thesis submitted in partial fulfillment of the requirements for a B.Sc. Degree in Geomatics

in the

Department of Geomatics

November 2015

Declaration of Authorship

I, JAYREN KADAMEN, declare that this thesis titled, 'Determining Scale within Unstructured Point Clouds' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for an undergraduate degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

EBE Faculty: Assessment of Ethics in Research Projects (Rev2)

Any person planning to undertake research in the Faculty of Engineering and the Built Environment at the University of Cape Town is required to complete this form before collecting or analysing data. When completed it should be submitted to the supervisor (where applicable) and from there to the Head of Department. If any of the questions below have been answered YES, and the applicant is NOT a fourth year student, the Head should forward this form for approval by the Faculty EIR committee: submit to Ms Zulpha Geyer (**Zulpha.Geyer@uct.ac.za**; Chem Eng Building, Ph 021 650 4791).
NB: A copy of this signed form must be included with the thesis/dissertation/report when it is submitted for examination

This form must only be completed once the most recent revision EBE EiR Handbook has been read.

Name of Principal Researcher/Student: Jayren Kadamen Department: Geomatic Engineering

Preferred email address of the applicant: jayrenparker@gmail.com

If a Student: Degree: BSc in Geomatics Supervisor: Dr. George Sithole

If a Research Contract indicate source of funding/sponsorship:

Research Project Title: Determing Scale for Unstructured Point Clouds

Overview of ethics issues in your research project:

Question 1: Is there a possibility that your research could cause harm to a third party (i.e. a person not involved in your project)?	YES	NO ✗
Question 2: Is your research making use of human subjects as sources of data? If your answer is YES, please complete Addendum 2.	YES	NO ✗
Question 3: Does your research involve the participation of or provision of services to communities? If your answer is YES, please complete Addendum 3.	YES	NO ✗
Question 4: If your research is sponsored, is there any potential for conflicts of interest? If your answer is YES, please complete Addendum 4.	YES	NO ✗

If you have answered YES to any of the above questions, please append a copy of your research proposal, as well as any interview schedules or questionnaires (Addendum 1) and please complete further addenda as appropriate. Ensure that you refer to the EiR Handbook to assist you in completing the documentation requirements for this form.

I hereby undertake to carry out my research in such a way that

- there is no apparent legal objection to the nature or the method of research; and
- the research will not compromise staff or students or the other responsibilities of the University;
- the stated objective will be achieved, and the findings will have a high degree of validity;
- limitations and alternative interpretations will be considered;
- the findings could be subject to peer review and publicly available; and
- I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism.

Signed by:

	Full name and signature	Date
Principal Researcher/Student:	Jayren Kadamen Jayren Kadamen	30/07/2015
This application is approved by:		
Supervisor (if applicable):	G. Sithole	17/8/2015
HOD (or delegated nominee): *Final authority for all assessments with NO to all questions and for all undergraduate research.*	J.Smit R Govind	5/8/2015
Chair : Faculty EIR Committee For applicants other than undergraduate students who have answered YES to any of the above questions.		

"Look up at the stars and not down at your feet. Try to make sense of what you see, and wonder about what makes the universe exist. Be curious."

Stephen Hawking

UNIVERSITY OF CAPE TOWN

Abstract

Faculty of Engineering and the Built Environment
Department of Geomatics

B.Sc. Degree in Geomatics

Determining Scale within Unstructured Point Clouds

by Jayren Kadamen

This research set out to develop a fully automated way for determining scale in a scene represented by an unstructured point cloud. Currently neither post-processed nor real-time generation of point cloud data provide real world scales without the need for a definite measurement to be known. A proof of concept test is proposed which tests a new method where an object with generalised dimensions is recognised and used to propagate scale throughout the scene. In contrast with existing methods, at no point is a manual measurement needed within the scene, nor will a calibration pattern or object with definite dimensions be needed for the purposes of object recognition.

In the test, the difference between generalised and actual dimensions per object class was assessed, and a maximum difference of 3.77% (2.8cm) was observed. The most notable finding was that a more reliable scale was provided by the ratio between objects which have dimensions determined by human anatomy (such as desks and chairs). This is in contrast to using the general dimensions of each object type to individually scale the scene. Thus better results were obtained by analysing the ratio between the heights of chair seats and desk tops using normalised values rather than using the absolute heights of these objects based on their reference value.

Recommendations are made for further testing such as analysing different types of indoor scenes and further development for both mobile and desktop platforms.

Keywords
Unstructured Point Cloud, Monocular Camera, Real-world Scale, Smartphones, CBIR, Object Recognition

Acknowledgements

I wish to express my deepest gratitude and appreciation to the following people who have contributed to the completion of this thesis:

First and foremost I would like to thank my project advisor Dr George Sithole not only for his guidance, patience, and invaluable insight during this thesis but throughout my journey through the degree as a whole. He has significantly changed the way I think, approach problems and fuelled my interest in this field of study.

Matthew Westaway for always being available to consult on problems and offer advice during the course of this thesis no matter how many times I bothered him regardless of the ridiculous hours we kept in the lab.

Laura Czerniewicz and Shonaig Harvey for proofreading my thesis in the extremely short time frame I gave them.

Dr Ramesh Govind for offering his guidance and insight as well as ensuring I continue to give my best.

The NSFAS and UCT's financial aid department without which I would not have been able to study at UCT.

Contents

List of Figures

List of Tables

Abbreviations

API	**A**pplication **P**rogramming **I**nterface
AR	**A**ugmented **R**eality
BRIEF	**B**inary **R**obust **I**ndependent **E**lementary **F**eatures
CBIR	**C**ontent-**B**ased **I**mage-**R**etrieval
CSV	**C**omma **S**pace **D**elimited
DARPA	**D**efense **A**nd **R**esearch **P**rojects **A**gency
EGS	**E**nvironmental & **G**eographical **S**cience
FAST	**F**eatures from **A**ccelerated **S**egment **T**est
FDN	**F**ixed **D**istance **N**eighbours
FLANN	**F**ast **L**ibrary for **A**pproximate **N**earest **N**eighbors
GPS	**G**lobal **P**ositioning **S**ystem
HSV	**H**ue **S**aturation **V**alue
IDE	**I**ntegrated **D**evelopment **E**nvironment
IMU	**I**nternal **M**easurement **U**nit
KNN	**K** **N**earest **N**eighbours
LiDAR	**Li**ght **D**etection **A**nd **R**anging
LSD-SLAM	**L**arge **S**cale **D**irect Monocular-**SLAM**
LS-SVM	**L**east **S**quare-**S**upport **V**ector **M**achines
MEMS	**M**icro**E**lectro**M**echanical **S**ystems
ORB	**O**rientated FAST and **R**otated **B**RIEF
OpenCV	**Open** Source **C**omputer **V**ision
PCA	**P**rincipal **C**omponents **A**nalysis
PCD	**P**oint **C**loud **D**ata
PCL	**P**oint **C**loud **L**ibrary
PnP	**P**erspective **n**-**P**oint

PTAM	**P**arallel **T**racking **A**nd **M**apping
RAM	**R**andom-**A**ccess **M**emory
RANSAC	**RAN**dom **SA**mple **C**onsensus
SfM	**S**tructure **f**rom **M**otion
SIFT	**S**cale **I**nvariant **F**eature **T**ransform
SLAM	**S**imultaneous **L**ocalization **A**nd **M**apping
SURF	**S**peeded **U**p **R**obust **F**eatures
VS	**V**isual **S**tudio
vSLAM	**V**isual **SLAM**
VTK	**V**isualization **T**ool**k**it
WiFi	**W**ireless **F**idelity

Chapter 1

Introduction

1.1 Background

In recent years acquiring a point cloud representation of a real-world scene by photogrammetric means has become increasingly popular. This can be attributed to advancements in multiple-image reconstruction algorithms, increased processing power of consumer-grade computers and the relatively inexpensive cost of cameras compared to laser scanners. Smartphones have placed a high quality imaging platform at the fingertips of billions of users (Richter, 2012). This has resulted in an increased volume of remotely sensed scenes using a single camera and the rise of online image sharing services such as flickr. These services can be searched by Content-Based Imge-Retrieval (CBIR) systems to access multiple images of the same scene for reconstruction purposes. There are also new platforms entering the market that make use of a single camera to capture a scene; these include commercial drones which have a growing market-share (Insider, 2015).

Photogrammetry is not the only field that makes use of point clouds. Computer vision has made use of unstructured point clouds for indoor mapping purposes. This is because cameras are chosen over active systems such as laser scanners because they require less power and are significantly less expensive. Robotic systems that use cameras for self-navigation often make use of stereo-camera systems where the baseline distance between the cameras is used to propagate scale throughout the resulting point cloud. However using a stereo camera system is not always possible. For example the platform may not

be large enough to position the cameras at a distance that allows adequate triangulation of common points so single camera (monocular) systems are used as an alternative.

1.2 Problem Statement

Real-world measurements are needed from Point Cloud Data (PCD) that has been formed by reconstructing multiple images from a single camera to represent a scene. The reconstruction process does not provide a real-world scale.

1.3 Research Objective

The objective of this report is to develop a proof-of-concept test for a fully automatic method of determining a non-arbitrary scale for unstructured PCD of an indoor scene.

1.4 Methodology

The research conducted took form in three stages:

- A literature review was conducted to research how cutting-edge technologies such as Augmented Reality (AR) solved the problem of obtaining a real-world scale for unstructured point clouds. It also details research into point cloud processing techniques that were used in the course of this report.
- A proof-of-concept test which uses object recognition to identify objects in a scene whose height value is expected to lie within a particular range. This height value could then be used to propagate a real-world scale throughout the scene for other objects whose dimensions are not known.
- The viability of the proof-of-concept test is assessed based on the difference between measured and reference heights. An analysis of variance test is performed in order to determine whether the heights for objects between rooms belonged to the same population.
- Lastly recommendations for further development are proposed in order to develop a solution that can be mass produced.

1.5 Expected Outcomes

To determine whether it is possible to develop a fully-automatic way of deriving scale for PCD using object recognition. The difference between the reference and measured heights of objects is not expected to exceed 10%.

1.6 Scope and Limitations

The scope of this research will be limited to deriving a scale for indoor scenes only. It will not feature a fully-realised solution but rather take the form of a proof-of-concept test which illustrates that scale for a scene can be obtained through fully-automatic means by leveraging object recognition. Each object will be recognised solely by their height which will fall within a particular range.

1.7 Outline of Report

The structure of the report will be as follows:

- **Chapter 2** will feature a literature review that is split into three parts. The first details sources of PCD whilst the second covers methods of scene reconstruction from multiple images and how a real-world scale is currently obtained. The third section will discuss cutting-edge technologies that have had to solve the problem of obtaining a real-world scale before ending with specific papers that have contributed significant research to the method proposed herein.
- **Chapter 3** covers the method used to demonstrate a proof-of-concept test to develop a fully automatic way of estimating a real-world scale for unstructured points clouds.
- **Chapter 4** discusses the results obtained using the method detailed in chapter 3.
- **Chapter 5** states the conclusions that were made concerning the results. It also details future recommendations for further development in order to offer a fully-realised solution.
- Lastly **Chapter 6** details recommendations to the proposed concept. Challenges and implications of developing smartphone and web-based applications are also discussed.

Chapter 2

Literature Review

This chapter serves to provide a background to how terrestrial PCD is obtained, why scale is arbitrary and how a real-world scale is derived for each acquisition method. Point-cloud processing has seen research predominantly within the field of computer-vision and as such the tools detailed in the following sections all originate from branches of this field. This can be attributed to the widespread use of cameras in robotic-based computer-vision problems due to their relative affordability and low power consumption in comparison to active sensors such as Light Detection And Ranging (LiDAR) systems. Current solutions that have successfully determined a real-world scale for a model or scene will be discussed ranging from manual to fully automatic methods.

The penetration of smart-phone devices will be discussed as they have the ability to capture scenes using images and video. Their growing and widespread use (Goldstein, 2014) has the potential to result in these devices becoming major contributors for remotely-sensed scenes using a monocular camera set-up.

Cutting-edge technologies which have had to solve the problem of obtaining a real-world scale for PCD such as AR systems will be covered by this chapter before ending with a discussion concerning specific papers which have contributed significant research to the proposed method detailed herein.

2.1 Sources of PCD

PCD can be obtained from a variety of sources from active sensors such as laser scanners and structured light systems to cameras (passive sensors). Laser scanners however are range measuring devices and as such contain a real-world scale (hence the need for calibration of these devices). Structured light systems such as the Kinect (Windows®) and David Laserscanner (DΛVID®) have a known base-length between the camera and projector which is used to propagate scale throughout the generated PCD. Monocular camera set-ups however are unable to generate PCD with a non-arbitrary scale without additional data.

The sections that follow detail methods of obtaining PCD for monocular set-ups for terrestrial scenes (compared to aerial photogrammetric methods).

2.1.1 Structure from Motion

The practice of determining the motion of the imaging platform (such as a camera) from the change in scene content (from photographs or video frames) is known as Structure from Motion (SfM) (Scaramuzza et al., 2009). Given multiple images of a finite number of fixed 3D points the task is to estimate the projection matrices and recover the 3D coordinate of X_j in figure 2.1 on the following page from the corresponding image points x_{1_j}, x_{2_j}, x_{3_j}. The problem however is that each pair of images (P_1 and P_2 in figure 2.1 for instance) have their own unique scale to allow for the vectors from the image points through the camera perspective centres to intersect at a common object point, (X_j in figure 2.1). Open Source Computer Vision (OpenCV) offers the Perspective n-Point (PnP) algorithm along with Random Sample Consensus (RANSAC) to solve for the position of subsequent cameras (P_3) using the object points that have already been identified (McCann, 2015). Scale is not determined due to any physical phenomenon but rather as a result of what is mathematically required in order to reconstruct a scene through triangulation of multiple points.

Due to this it is well documented within SfM that obtaining a real-world scale can only be achieved when one knows the baseline distance between subsequent frames of the camera motion (from P_1 to P_2 and P_3 in figure 2.1) or an element in the scene must have a known real-world dimension (Fergus, 2012).

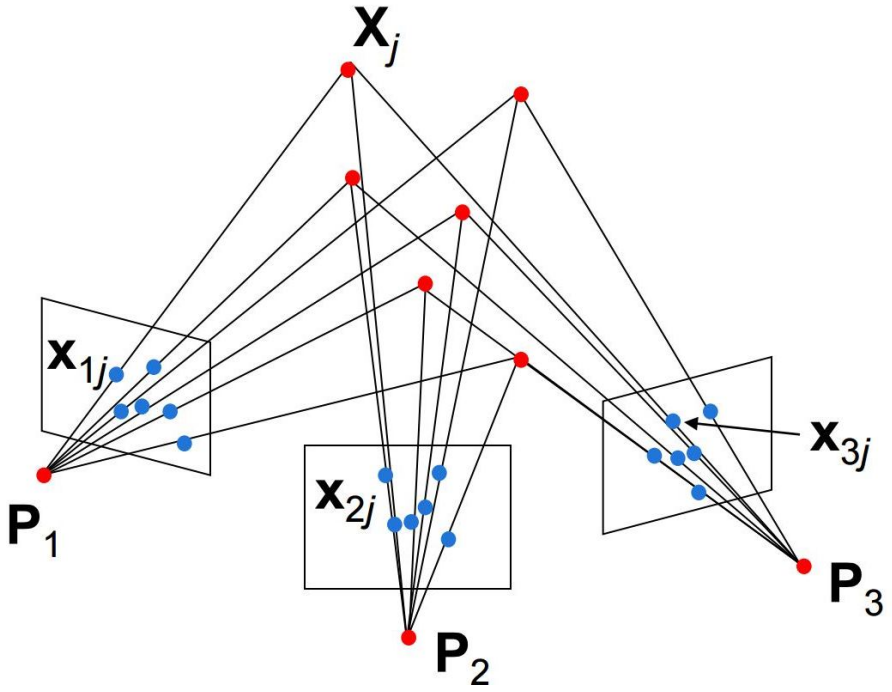


FIGURE 2.1: 3 images P_1 to P_3 all contain the real-world 3D point X_j which has corresponding image coordinates $x_{1_j}, x_{2_j}, x_{3_j}$. These image points are used to align the images such that X_j is reprojected to a common position in 3D space with an arbitrary scale (Fergus, 2012).

Obtaining the baseline distance can be achieved by making use of a combination of Internal Measurement Unit (IMU) and Global Positioning System (GPS) systems such as the one proposed by Ntzi et al. (2011). The use of these additional technologies is not always readily available especially with images and video that have already been captured.

2.1.2 Multi-View Stereo

The primary goal of multi-view stereo is to generate a dense 3D model from multiple views of an object or scene (Seitz et al., 2006). 'Scene flow' is an expansion of this which includes scenes that have been captured using imaging platforms that move in a non-rigid manner such as video (Vedula et al., 2005).

Points from the same object within a scene captured from multiple images need to be matched and anchored to the same physical point. This is a difficult task as multiple views of the same scene often result in objects being subject to occlusion between views, perceived changes in shape and alteration of appearance. The procedure to recreate the 3D scene from 2D images measures whether the "3D model is consistent with the

input images" (McCann, 2015). Parameters such as colour, edges and texture can be used to measure consistency within the collection of images that offer multiple views of a scene or object. Two approaches are used to create the final 3D model the first requires many views as it builds the model up from points which can be seen throughout a large percentages of the input images. The second requires high resolution imagery in order to capture significant texture information and good geometry as it starts with a bounding box and inconsistent points are eliminated from the model.

Transforming images to fit features within the scene can lead to transformations errors from rotations and scaling. The Scale Invariant Feature Transform (SIFT) algorithm uses multiple techniques to provide scale, rotation, illumination and small view-point change invariance. Speeded Up Robust Features (SURF) is partially based upon, and can be seen as an evolution of, the SIFT algorithm. It boasts faster computation times and is more robust at mitigating transformation errors (Bay et al., 2006). Figure 2.2 illustrates SURF matching given two views of a Rubiks cube.

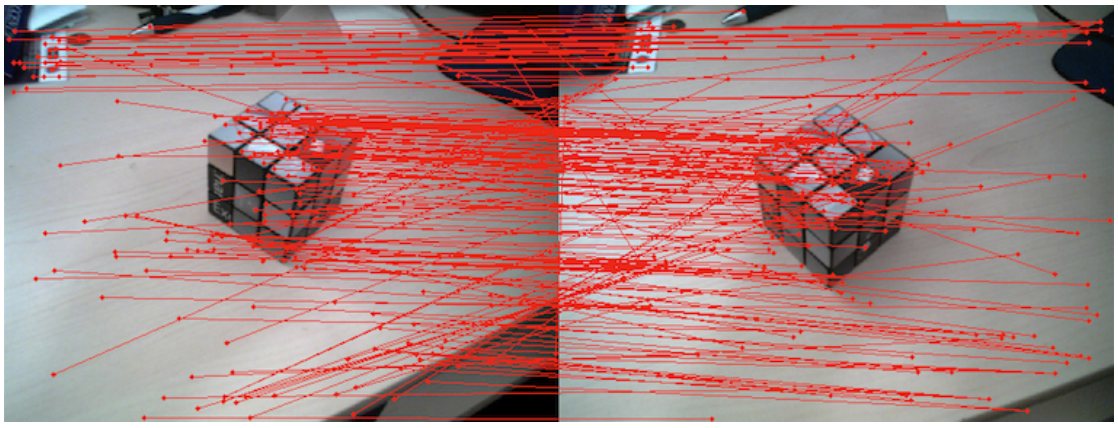


FIGURE 2.2: Red lines identify matching points using the SURF algorithm between two images with different vantage points of the same Rubiks cube (Anon., 2011).

The process of forming the 3D model involves stitching together multiple images in order to recreate the scene based upon features. This is done to create a 3D model that is consistent with the given images rather than based on a physical phenomenon such as accurately re-projecting the bundle of rays that created each image originally. As such this method results in a model with an arbitrary scale.

2.2 Obtaining a Real-World Scale for PCD

2.2.1 Manual Measurements

Physically measuring a dimension in the scene and propagating it through the generated 3D model is an established method of providing a real-world scale for PCD. This can be accomplished by measuring an item in the scene or by using targets with known coordinates in a 3D reference frame. An example of these targets can be seen below in figure 2.3. They are distinctly coloured black and white so that they are visually distinguishable in a scene.

An example of a terrestrial target used in both laser scanning (to tie multiple scans together) and photogrammetry (for the purposes described in this chapter) can be seen below in figures 2.3 & 2.4.

FIGURE 2.3: Black and White Targets for Photogrammetry and Laser Scanning are indexed or their coordinates are known and used to tie multiple images or scans together. (Leica Geosystems HDS Targets)

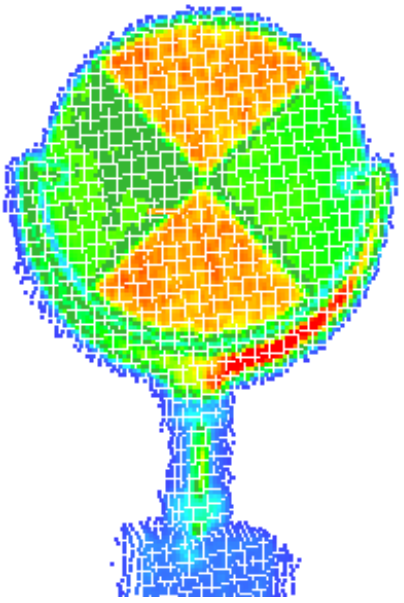


FIGURE 2.4: False colour representation of the same target on the left based on intensity values from a laser scanner. This makes it visually distinct and easier to identify during the post-processing phase where scans are tied together. (Registration, Best Fit, Alignment)

Coordinates of these targets are ascertained through surveying means such as a traverse. The targets are then tied to a real-world coordinate system and are then identified in the PCD to be used as control points. The distances between them provide a non-arbitrary scale as it defines the relationship between points through real-world distances thus creating a metric space.

Making use of manual measurements introduce an element of human error. This can take the form of insufficient measurements to offer adequate geometry or to distribute error through the scene. With respect to measuring a single dimension in the scene there is a risk that the measurement can be read incorrectly, forgotten to be made or documented value could be lost. Manual measurements require additional effort, equipment and time to perform and for point clouds that have already been captured they often do not offer a solution to obtaining a real-world scale.

2.2.2 Use of Known Objects

Object detection is used to find instances of real-world objects such as people, cars or bicycles in scenes that originate from video or images. Objects are described using descriptors called features which can take the form of texture, colour and many other attributes. A unique collection of these features create a feature space which describes an object (Davidson-Pilon, 2012). Objects with known features are stored in a database and algorithms search the scene to find collections of features which match those stored in the database. This can be seen in figure 2.5 below where items commonly found in a kitchen are attempted to be identified by a program written with the OpenCV library.

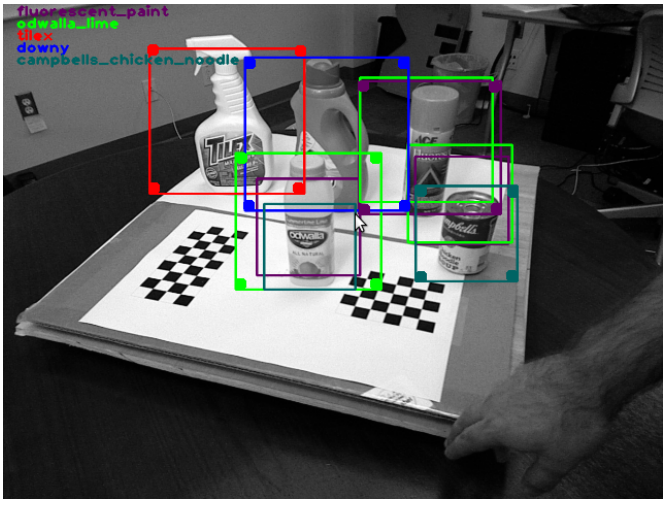


FIGURE 2.5: Object detection identifies the checkerboard pattern which is used to obtain a scale for the scene as the size of the squares are known. The absolute scale is used to identify other objects where measurements are descriptors (DevZone, 2011).

Object detection attempts to solve the problem of obtaining a non-arbitrary scale for PCD by placing objects which have at least one feature that describes a known dimension within a scene. In the above figure a calibration pattern (checkerboard) serves as this type of known object as each black and white square has a known dimension. Using SfM the distance between subsequent frames of the video feed containing the above scene can be recovered and used to propagate scale throughout the resulting PCD (Fleet et al., 2014). The calibration pattern can also be used to determine the interior parameters of the camera.

Once an absolute scale has been determined the other objects within the scene such as Campbell's can of soup can be described by its true height. In figure 2.5 above bounding boxes with squares at the corners depict an object with a high match against its descriptors in the database whilst other boxes represent partial matches.

Example of Using Object Detection to Derive Absolute-Scale

The paper presented by Rashidi et al. (2014) details a method of obtaining real-world measurements from PCD that has been generated by monocular photography and videogrammetry. The problem presented with the current solutions is that it requires manual measurements (detailed in chapter 2.2.1) to obtain real-world scale.

The solution proposed by this paper was to make use of object detection with specific objects for outdoor and indoor scenes. These objects came in the form of an $A4$ piece of paper for indoor scenes and a cube with known dimensions and distinctly coloured sides for outdoor scenes. The $A4$ sheet of paper was chosen as it is a commonly found item in indoor environments thus making it readily available with dimensions already known. The algorithm used is able to extract the corner points of both the cube and the $A4$ sheet in order to match them in subsequent frames. These objects are then used to determine a non-arbitrary scale. The method and results of the indoor scenes will be discussed.

The $A4$ sheet requires the four corner points to be detected, as a result epipolar geometry was used to identify the corresponding points in the second view based on the assumption that the corner points follow a clockwise order. The four corners were detected by first identifying the page itself (by filtering the Hue Saturation Values (HSV) of the scene). A

modified Hough transform, to account for lines appearing curved due to lens distortions, is used to identify the edges of the sheet. The edges are extended until they intersect the neighbouring edge at a point thus providing the four corners of the $A4$ sheet.

To test the performance of the above concept a number of scenes were captured by means of a video taken by an off-the-shelf video camera. Each scene was captured as completely as possible in order to minimise occlusions. In order to determine the discrepancy between true distances and those obtained from the generated point cloud manual measurements were made with a Leica Laser Disto. The average length measurement error for indoor scenes were $0.14cm/m$. The results offer promising accuracies using a medium such as video that suffers from blurring between frames, lower resolution than photography and varying principal distance (due to autofocus).

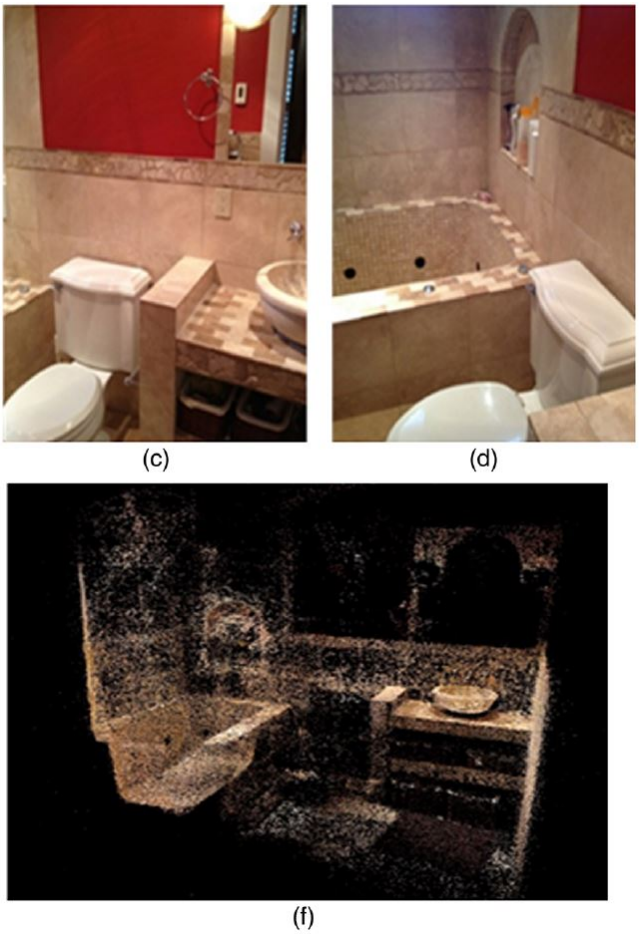


FIGURE 2.6: **Reconstructed Bathroom from Video:** *(c)* and *(d)* illustrate vantage points of an indoor bathroom scene from frames of a video whilst *(f)* illustrates the resulting reconstructed point cloud from a monocular set-up (Rashidi et al., 2014).

The shortcoming with the method discussed above is that a known object needs to be placed in the scene before it can be captured. If the cube or $A4$ paper is not readily

available the absolute-scale for the scene cannot be determined using this method. If the cube experienced physical damage during transportation or the $A4$ paper was not perfectly straight (or taken from an exam pad where the top strip has been trimmed off altering the dimension of it) the consequence would be an incorrect scale for the resulting PCD of the scene.

Using object detection to obtain a real-world scale for PCD can be seen as a semi-automatic method because a prepared object with known dimensions must be placed in the scene beforehand. Even so it can be seen as an improvement over manual measurements in terms of equipment required and sources of error described in chapter 2.2.1.

2.2.3 Content-Based Image Retrieval

The field of computer vision has seen a sharp rise in research within CBIR which attempts to describe scenes or images based on the content in the image rather than relying on metadata alone. The rise in research can be attributed to the incorporation of high quality imaging sensors becoming common-place in smart-phone devices. The latter along with internet services that offer image sharing has led to a rise in the sheer volume of imagery readily available over the internet thus resulting in a growing need for improved methods of querying such a rapidly expanding database (Yeh et al., 2004).

The "content" referred to within images are called features, a collection of which are used to describe an object (which has the same definition as that outlined in chapter 2.2.2). Describing the content of an image is not the primary goal of CBIR systems but rather to generalise the description of content within an image or PCD. This allows the description to be used in a variety of applications from object recognition to reverse image searches where the quality of the result (matching object or image respectively) can be assessed.

One of the tools of CBIR is object recognition, an extremely important component of computer vision systems. The primary goal of this tool in relation to robotics is to give robots equipped with imaging sensors the ability to recognise unknown objects within a scene. Before the images are re-projected to create a 3D scene the objects within the images can be recognised in order to ascertain further descriptors for it once its 3D

counterpart has been generated. This method can draw from a larger image database which is remotely accessed via the internet in order to select the appropriate 3D object stored in a local database. Additional adjacent features from the images can also be used to better locate the object in the 3D scene from that recognised in the images. This method uses a hybrid of server and client based systems as well as 2D and 3D object recognition.

In figure 2.7 objects are recognised based on geometric shape. The descriptors need to be specific enough not to confuse the blue and red toy car in the top centre with the maroon hole-punch (top left) but also general enough to be able to detect any type of hole-punch regardless of colour for example.

FIGURE 2.7: Even under non-uniform lighting conditions and occlusions the object recognition system illustrated above can still identify objects successfully. Each object type has a different colour bounding box, (the shoe has a green box whilst the punch has a dark blue box) (Bormann and Hampp, 2014).

Once an object has been recognised based on its geometric features it can also be classified by comparing them against all the classes within its database to see which class offers the best fit (Lai et al., 2013). This allows a computer vision system to ascertain which object is in the image rather than where a particular object is in an image as in the case of object detection (Penelope, 2013).

The same criteria to obtain a real-world measurement using object detection is present here in object recognition (at least one descriptor of an identified object must be a measurement). The benefit however of using object recognition is that it can identify many possible objects within a scene rather than having to place a known object such as a calibration pattern within it beforehand. This allows the scale of PCD, that has already been acquired, to be determined.

2.3 Discussion

2.3.1 Augmented Reality

AR "is a technology that superimposes a computer-generated image on a user's view of the real world, thus providing a composite view" (Sadiku and Ali, 2015). These systems need to be capable of locating objects in a real-world environment and aligning geometric models to them (Whitaker et al., 1995). An example of this can be seen in the figures below where an AR system recognises a MathWorks magnet on the desk (figure 2.8) and overlays a video on top of it in real-time (figure 2.9).

FIGURE 2.8: AR system using object recognition to identify the MathWorks fridge magnet.

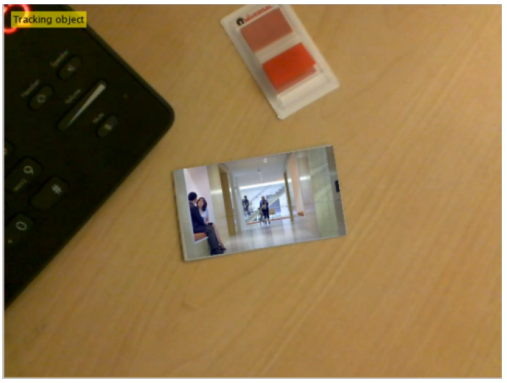


FIGURE 2.9: AR system overlays a video onto the recognised object in real-time.

Object Recognition and Tracking for Augmented Reality

A non-arbitrary scale is required in order to position and align the supplementary computer-generated imagery onto the relevant real-world object. Established methods of achieving a real-world scale make use of object detection in the form of calibration patterns or known objects but these are not always available. Modern methods include the use of object recognition such as the system proposed by Lepetit et al. (2003) where an image of the scene is registered online by matching feature points against a database. In this solution the emphasis was placed on using a server based approach to alleviate the computational burden so that the client can concentrate on delivering the computational performance required to run the AR system.

To resolve the internal and external orientation parameters object recognition or detection methods can be used. These can also be used to derive a real-world scale if a dimension of the detected or recognised object is known. Calibration drift arises when the camera moves and this accumulates once the object used for calibration is no longer in the field of view. This results in a need to re-calibrate at set intervals, often based on the algorithm used and the level of precision desired. This can be solved by returning to the position where the calibration object is located or by placing these objects along a pre-determined path at intervals where re-calibration is needed to mitigate the effects of drift. Each calibration calculation is processor intensive as it uses a least squares bundle adjustment (Tsai (1987) for instance or Kalman filtering (Mirzaei and Roumeliotis, 2008)).

In the field of computer vision a commonly occurring goal is to create a platform that can navigate and explore a scene autonomously. This often results in an imaging platform that is battery powered which limits the processing power available to the client as performance is traded for energy-efficiency. The calibration calculation can be computed on the server and sent back to the client but this relies on having an internet connection that is fast enough for this to be the most efficient solution.

2.3.2 Simultaneous Localization and Mapping

Simultaneous Localization and Mapping (SLAM) modules are tasked with creating and updating a virtual environment that represents an unknown real-world scene. This has to occur in real-time whilst the scene is explored and the location of this device must be constantly tracked. Modern iterations of SLAM based systems feature multiple sensors each with a strength related to their function; LiDAR offers range measurements whilst cameras contribute colour. Examples of modern autonomous SLAM based systems can be found in the Defense and Research Projects Agency (DARPA) Robotics Challenge which has set the standard in this field (Molinos et al., 2014). Even though solutions to the DARPA Robotics Challenge rarely rely on monocular camera set-ups for real-time mapping the equipment used such as LiDAR is expensive and power-hungry putting them out of reach for small scale robotic applications. Cameras are more accessible from a financial standpoint and have comparatively low power requirements in their passive form yet they need to be calibrated using the methods discussed in chapters

2.2.2 and 2.2.3 which are subject to the drawbacks detailed in chapter 2.3.1. Visual SLAM (vSLAM) makes use of a monocular camera rather than a variety of sensors to solve the problem of localization and mapping by generating PCD to form a 3D representation of the scene (Weiss et al., 2014).

SfM has matured to the point where post-processing a scene captured by multiple images from a single camera in order to recover the route traversed is procedural rather than problematic (Fitzgibbon and Zisserman, 1998). vSLAM however requires this to be determined in real-time which introduces challenges not faced by post-processed solutions. An example problem is the inability to distribute error evenly through the captured images as they cannot be batch processed simultaneously because new regions of the scene are constantly being incorporated. The real-time constraint forces vSLAM modules to be efficient. If the images are received at a constant rate such as from a camera at $25Hz$ the computations must operate in constant time and as such the processing time per image cannot increase endlessly otherwise the real-time constraint will be breached (Davison, 2003). Mouragnon et al. (2006) presented the first real-time example of a bundle adjustment followed by Klein and Murray (2007) which presented Parallel Tracking And Mapping (PTAM) for AR using SLAM.

Initial calibration is performed using object detection with early attempts at combating drift making use of short-lived features akin to those used in post-processing methods (Ayache, 1991, Beardsley et al., 1995, Harris, 1992). Mur-Artal et al. (2015) represents a modern method of solving the problem by using Orientated FAST and Rotated BRIEF (ORB), a computationally efficient and robust local feature detector which draws from the Features from Accelerated Segment Test (FAST) feature point extraction algorithm and Binary Robust Independent Elementary Features (BRIEF). Large Scale Direct Monocular-SLAM (LSD-SLAM) developed by Engel et al. (2014) has relative-scale and the beneficial property of being able to distinguish near-field objects from those further afield allowing for better selection of object windows during a constantly changing view. The work by Pillai and Leonard (2015) incorporates both ORB and LSD-SLAM for object-recognition and uses the work of Strasdat et al. (2010) to combat scale, rotation and translation drift at loop closure. The system presented is able to access the map of its surroundings that it is currently observing whilst it builds new areas and the location of the camera is known at any point in time. This is referred to as a SLAM-aware system compared to a SLAM-oblivious one where objects are detected

and recognised as new frames are recorded without being aware of its location in the map that is being built (Pillai and Leonard, 2015). The benefit of this system is that it does not need to be recalibrated at regular intervals, it relies on returning to the initial position to distribute the error over the traversed path. The drawback is the need to return to the initial position.

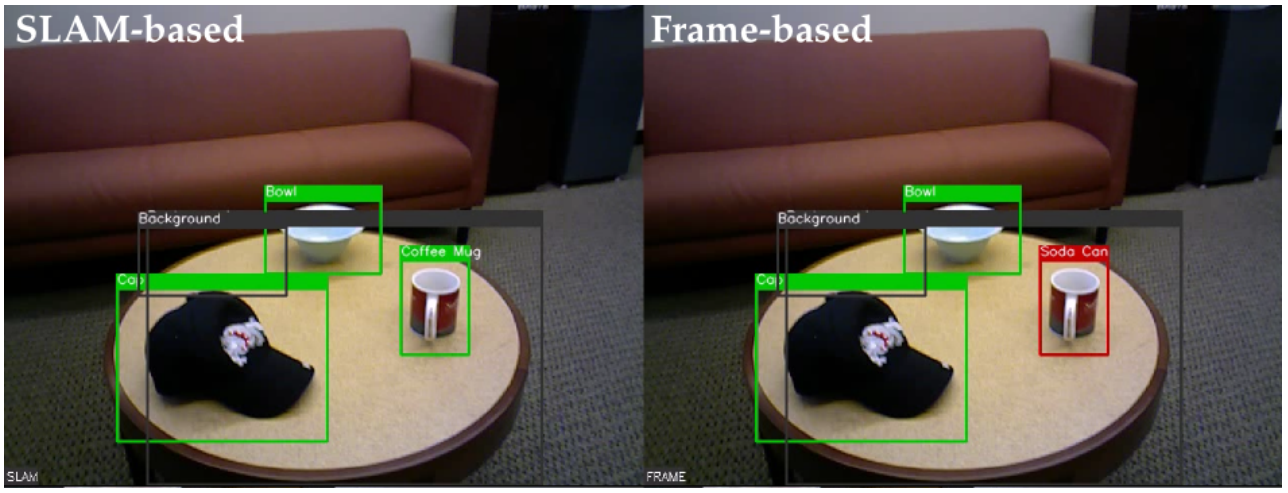


FIGURE 2.10: Comparison of the performance of SLAM-aware (left) versus SLAM-oblivious (right) object recognition where ambiguity is present. In the right image the coffee mug is misidenitified as a soda can and has a red bounding-box to highlight this misidentification (Pillai and Leonard, 2015).

2.3.3 Mobile Devices

Mobile devices such as smart-phones are becoming increasingly commonplace (Goldstein, 2014) and powerful in terms of processing power (Ye, 2015). They are also packed with MicroElectroMechanical Systems (MEMS) which feature a variety of technologies from GPS to mobile telecommunications. Smart-phones are currently being used for outdoor navigational purposes relying on a mixture of GPS and an internet connection for location. Once the device moves indoors however GPS is unable to provide a reliable location due to lack of reception. Mobile telecommunications can also be rendered inoperative once inside large structures or underground and WiFi may not be freely available. The IMU is able to track the path travelled but it is subject to drift which is cumulative thus rendering this technology imprecise in practice without GPS to rectify it (David Sachs, 2010).

Using a smart-phone for indoor mapping and localisation therefore presents challenges not encountered for outdoor navigation and localisation. Pei et al. (2012) presented a solution to the location problem by recording built-in accelerometers and magnetometers within smart-phones in various states of motion (motionless, whilst the user is standing or walking etc.). Least Squares-Support Vector Machines (LS-SVM) was then used to ascertain which motion was being performed in order to assist the wireless positioning algorithm that made use of WiFi signal strength to calculate the relative change from the rest position. The availability of WiFi in commercial areas has risen in recent years (Wakefield and News, 2014) making this solution relevant but it is still a power-hungry resource, a limiting factor for smart-phone devices which run off batteries.

Mostofi et al. (2014) presented a solution to the mapping and localisation problem with their SLAM based system which uses a monocular camera and the built-in IMU found on smart-phones. The system delays key-point initialisation to first correct the drift from the IMU and SLAM systems using the Extended Kalman Filter technique. The SLAM module uses the SURF algorithm assisted by RANSAC to recognise features and ascertain location and relative motion based on features already recorded. All of the calculations occur on the device rather than rely on wireless technologies to be available to communicate with remote processing services. The camera is not calibrated beforehand so it has an arbitrary and constantly changing scale. This can be resolved by using object-detection if a client-based solution is desired or object-recognition if a larger database of objects is required which may result in a hybrid solution of a server-based database and client-based processing.

2.3.4 Indoor Scene Analysis for Point Clouds

Objects can be described not only by local features but also by relationships it has within the scene itself. With scene analysis the arrangement of objects in relation to one another can be used as a descriptor (i.e. all computer screens are on desks in a classroom). This opens up a variety of descriptors based on the relationships between other objects such as relative distances, sizes and orientation. The first step is segmenting the collection of points that populate a cloud into objects based on their fundamental features.

Segmentation identifies collections of points and groups them into segments for further processing. For example a point cloud of a classroom will have four walls, a ceiling, a

floor, desks and chairs. In basic terms these objects are comprised of horizontal and vertical planes of varying sizes. A good segmentation model will be able to separate the smallest object (a single chair) from the largest object (floor or ceiling). This however depends on ones geometric understanding of the scene and for an algorithm to be robust it must incorporate a level of generalisation. There are a number of models which Point Cloud Library (PCL) offers for this but for the purposes of this thesis only region-growing segmentation will be discussed.

Segmentation of Point Clouds using Smoothness Constraint

An integral part of automatic point cloud processing is segmentation. The paper by Rabbani et al. (2006) aimed to offer an alternative to curvature-based segmentation as it often led to over-segmentation resulting in the need for manual editing. Instead a smoothness constraint was used to segment the point cloud along with surface normals which allowed the algorithm to be robust.

The segmentation is broken up into two major steps, normal estimation and region growing. The normal to each point is estimated by making use of plane fitting where the size of the neighbourhood of points is specified by KNN or FDN. The residuals from plane fitting can be attributed to noise or a possible descriptor of high curvature as proposed by Rabbani et al. (2006).

The region growing algorithm used takes the point-normals along with the residuals and groups collections of points which have smooth-surfaces. The grouping is subject to the constraint that the points must be locally connected (making use of k-Nearest Neighbours (KNN) or Fixed Distance Neighbours (FDN)) and form a smooth surface (low variation of point normals).

The residual threshold, r_{th}, dictates the degree of under-segmentation or over-segmentation. Figure 2.11 on the following page illustrates the effect of adjusting the residual threshold of the resulting segmentation for one of four industrial scenes that were used. By changing the threshold between the normal of a selected seed point and its neighbours, θ_{th} the smoothness constraint can be controlled as well.

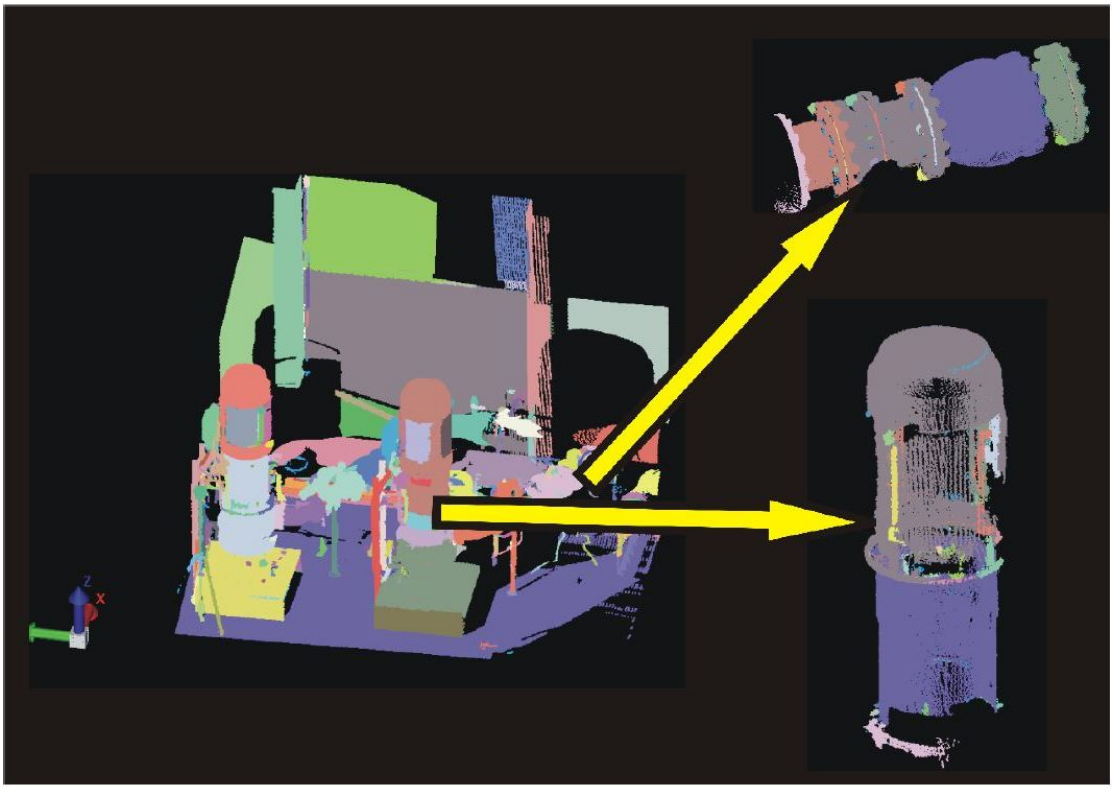


FIGURE 2.11: In the top right the object is comprised of large segments whilst the bottom right has individual strips for the cooling fins of the machine. This is achieved by adjusting the residual threshold to allow more segments (Rabbani et al., 2006)

The end result was a robust algorithm that could group smooth areas whilst giving the user a choice between under-segmentation or over-segmentation.

Relevance: Solving the problem of controlling under-segmentation and over-segmentation allows one of the major initial steps of point cloud processing to be fully autonomous - i.e. no need for manual editing to correct poor segmentation. If this can be accomplished then an in-depth understanding of the mathematical underpinnings of the segmentation need not be known to the end-user.

2.3.5 Object Recognition for Indoor Point Clouds

The field of object recognition within computer vision has experienced growing interest due to advances in energy-efficient processors on mobile platforms and lightweight and affordable cameras. This allows for high-quality camera's to be attached to mobile robotic platforms which requires powerful yet energy-efficient processors. The ultimate goal is to imbue the complex human vision system upon a robot. On the path to

attaining this computer vision systems are currently able to recognise objects in a scene based on the unique collection of features that describe it.

In chapter 2.3.4 the focus was on analysing the scene as a whole and whether adequate segmentation had taken place. The following paper by Rabbani et al. (2006) concentrates on recognising items within a scene for the purpose of attributing meaningful labels to them (for the purposes of robotic interaction).

Deriving Object Based Maps from PCD

Rusu et al. (2008) put forth a scenario where an autonomous robot is tasked with helping a human perform chores within a kitchen in an assisted living situation. The issues raised by this scenario are not just mapping the environment for the purposes of navigation but to recognise objects such as cupboards and appliances. The robot will have to know what items are stored within these objects or whether the object performs a specific function (such as a washing machine) so that it knows where to look in order to complete a specified task such as washing clothes.

The point cloud is converted into an environment object model where the robot is able to tell which objects are obstacles for navigation purposes and which areas of the scene can be fit to a model in order to recognise objects that are required to perform tasks.

Once the point cloud is acquired only geometric information concerning the scene is available which is adequate for position and navigation purposes but not for interacting with objects within the environment. For this to take place semantic information must be extracted which is done by the functional mapping module which makes use of object recognition to identify objects based on geometric descriptors.

The aforementioned module uses the same region-growing approach developed by Rabbani et al. (2006) and discussed in chapter 2.3.4 to segment the point cloud into recognisable planar surfaces. Once the point cloud has been segmented a model fitting algorithm uses a collection of cuboids, circles, and lines to fit cupboards, knobs, and doors respectively. An illustration of how the cuboids are fit can be found in figure 2.12 on the following page.

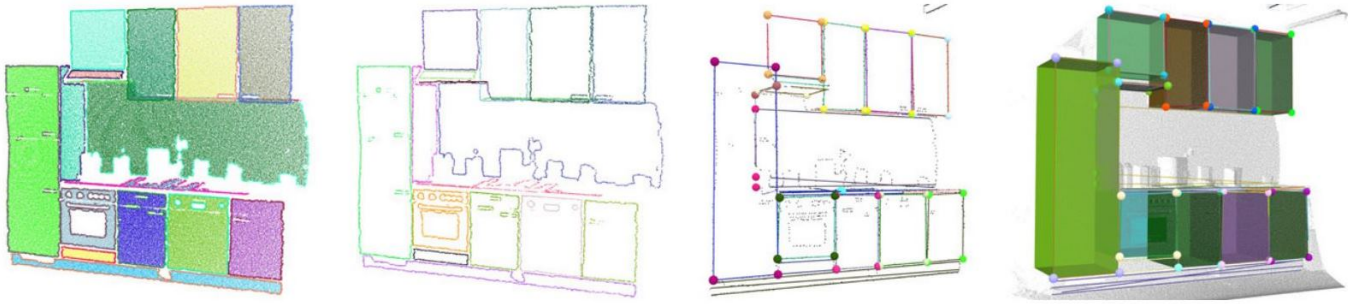


FIGURE 2.12: **From left:** Initial segmentation identifies planes (far left to centre left). Boundary points of each segmented region (centre right) are used to create best fitted lines to each boundary. Cuboids (far right) are created by projecting the 2D quad formed by the boundary points and best fitted lines back onto the wall. (Rusu et al., 2008)

The result was a map creation process that could take a point cloud which only offered an occluded view of an environment and apply semantic labels to recognised objects. This was accomplished so that a semantic map could be built allowing the robot to intelligently interact with the environment.

Relevance: The realisation that man-made indoor spaces consist of a collection of horizontal and vertical planes allows for intelligent segmentation. The use of shapes along with planes (cuboids and horizontal surfaces respectively) allow for contrasting of similar objects along with accurate attribution of their semantic labels (such as drawers and cupboards both have planar surfaces but can be told apart based on relative cuboid size and handle shape).

Chapter 3

Method

This chapter will discuss the steps followed to demonstrate a proof-of-concept test for the development of a fully automatic way of estimating a real-world scale for unstructured point clouds. The upcoming sections discuss in detail the concept, method and execution.

3.1 Concept

Indoor scenes such as office spaces, university lecture rooms, and household environments feature objects which are commonly used such as chairs, desks or tables. These objects have dimensions which can be estimated without the need for physical measurement. Table 3.1 below and figure 3.1 on the following page illustrates the average dimensions for commonly found objects:

Typical Measurements for Furniture and Doors		
Item	Dimension	Label
Chair	45-53cm	Seat Height
	40cm	Seat Length
	81-107cm	Backrest Height
Desk	74cm	Tabletop Height
	122-152cm	Tabletop Length
	74cm	Tabletop Width
Door	200cm	Height
	96cm	Doorknob Height
	61-91cm	Width

TABLE 3.1: Table showing the average dimensions of furniture and doors, (Lefler, 2004) and (Griggs, 2001)

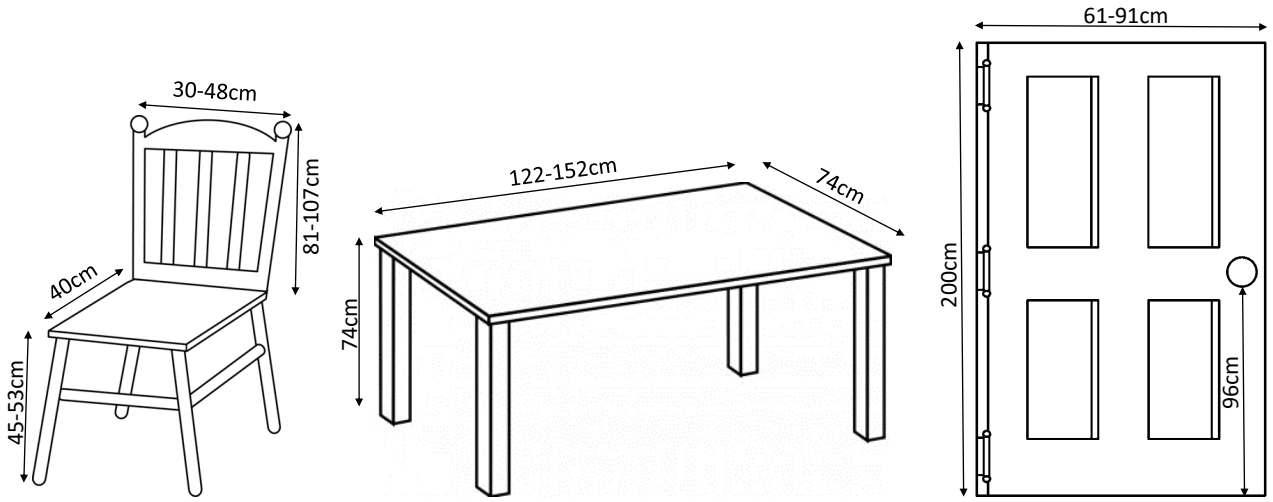


FIGURE 3.1: Schematic diagram of commonly found furniture (chairs, desks or tables, and doors) based on values from Lefler (2004) and Griggs (2001)

Using objects with dimensions that can be estimated provides a fully automatic way of determining scale for a scene by utilising object recognition. This means that no physical measurements need to take place within the scene or for the known object. This also allows for the scale of a scene that has already been captured to be determined.

The method that was derived to test the above hypothesis took the form of a proof-of-concept test. This meant that a fully-fledged CBIR system leveraging object recognition did not need to be developed. This fortunately meant that a database of known objects need not be developed either.

The following sections detail the steps outlined to test the above hypothesis from data capture to PCD processing and statistical testing. Indoor spaces on UCT campus were selected as test candidates. The types of rooms included a modern classroom within the newly constructed Snape Building, (Snape 3C), a meeting room (Environmental & Geographical Science (EGS) Seminar Room) and the Geomatics Postgraduate Computer Lab. A point cloud representation of each room can be found in figures 3.2, 3.4, and 3.3 on the pages that follow.

Each room contained different types of chairs with varying heights from older office chairs with wheeled bases (EGS Seminar Room) to modern ergonomic chairs (Snape 3C). Desks and tables also varied between rooms as the Geomatics Postgraduate Computer Lab featured cubicle desks with a large office table whilst the others contained a variety of tables.

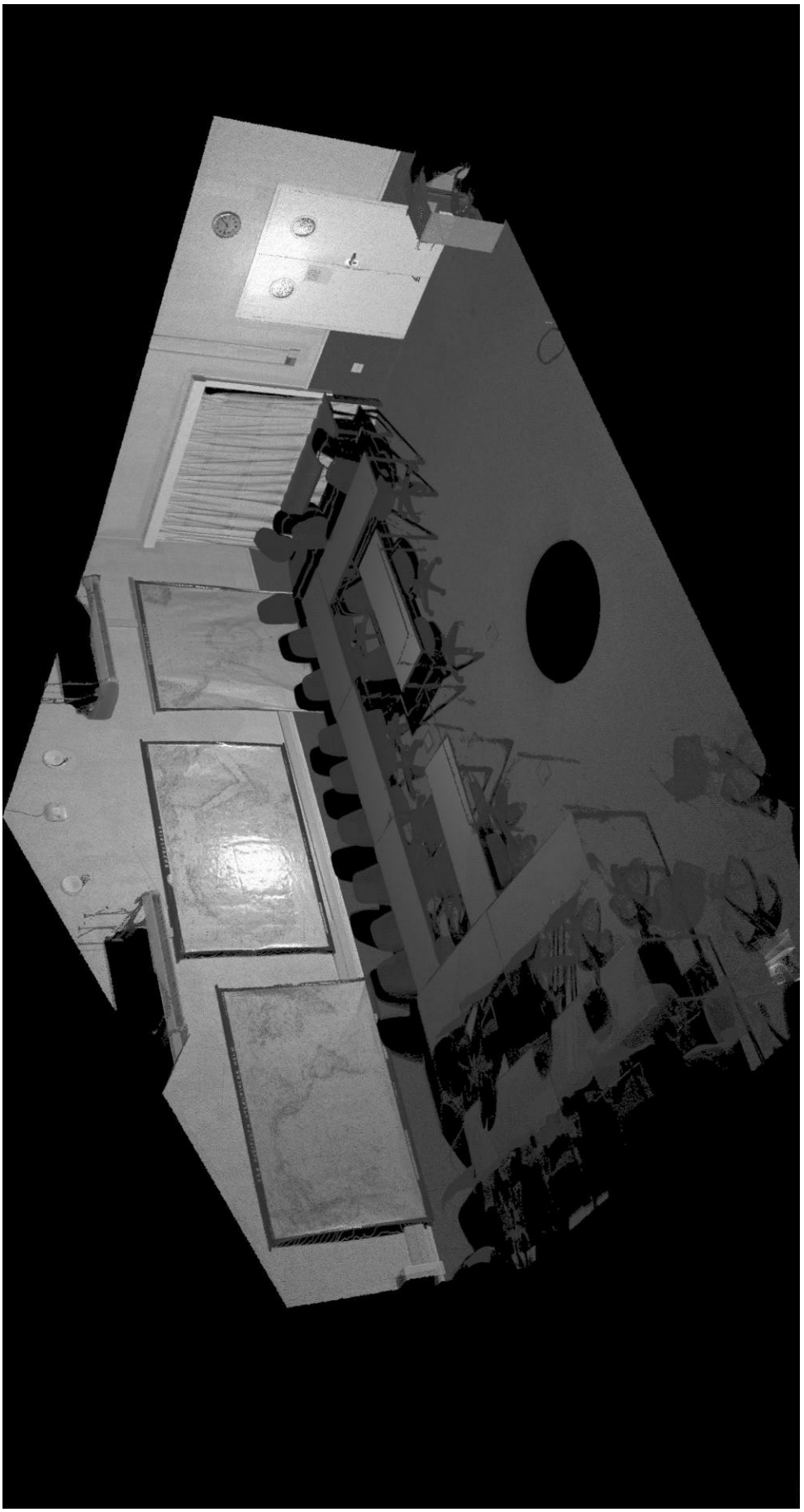


FIGURE 3.2: A point cloud representation of the EGS Seminar Room.

FIGURE 3.3: A point cloud representation of the Geomatics Postgrad Lab.

FIGURE 3.4: A point cloud representation of the newly constructred Snape 3C classroom.

3.1.1 Scanning and Pre-Processing Procedure

The Geomatics Postgraduate Computing Lab was scanned with the scanner placed directly atop a table. For Snape 3C and the EGS Seminar Room the scanner was mounted on a tripod to provide a less occluded view. Multiple set-ups per room were not performed even though this would have provided a more complete point cloud as this was to include occlusions in order to test the robustness of identifying objects solely on their relative height.

The PCD was cleaned to remove outlying points. These were points that had strayed outside of the indoor space. Lastly to reduce computation time the point clouds were thinned to a $1cm$ resolution. This limits the precision to which the dimensions of an object recognised within the room to $1cm$ as well. However this is not a cause for concern at the proof-of-concept stage.

3.2 Processing PCD

The program structure is outlined in figure 3.5 below and detailed in the upcoming subsections.

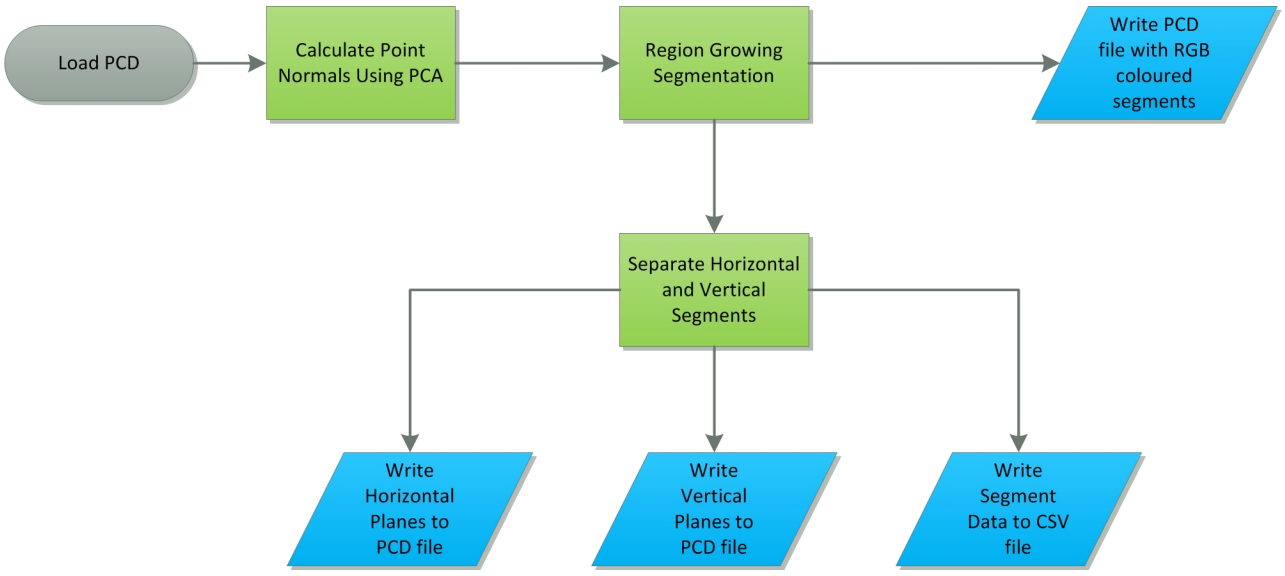


FIGURE 3.5: Program Workflow Diagram

3.2.1 Point Normals

The region growing algorithm from PCL was used to perform the segmentation. This algorithm makes use of the normal to each point in the supplied PCD and as such the first step is to calculate the point normals for the input PCD. This problem is solved using a built-in class within PCL called Normal Estimation. The implementation of this can be seen in figure 3.6 below.

```
// Calling normal_estimator function
pcl::NormalEstimation<PointType, pcl::Normal> normal_estimator;
// Setting search method
normal_estimator.setSearchMethod(tree);
// Providing input PCD
normal_estimator.setInputCloud(cloud);
// Setting the neighbour selection algorithm as KNN
normal_estimator.setKSearch(m_knn);
// Store the result in the variable 'normalcloud'
normal_estimator.compute(*normalcloud);
```

FIGURE 3.6: Implementation of the built-in NormalEstimation class within PCL

The point normal for each point is approximated by calculating the normal of a plane that lies tangent to the surface and passes through the seed point. This reduces the problem to a Principal Components Analysis (PCA) calculation and an analysis of the resulting covariance matrix, C. In equation 3.1 below N_i is the seed point, k is the number of nearest neighbours closest to the seed point, N_0 is the centre of the nearest neighbours (in 3D coordinate space), λ_j and $\vec{v}_j$ represent the $j-$th eigenvalue of the covariance matrix and the $j-$th value of the eigenvector respectively.

$$C = \frac{1}{k}\sum_{i=1}^{k}.(N_i - N_0).(N_i - N_0)^T, C.\vec{v}_j = \lambda_j.\vec{v}_j, j \in \{0, 1, 2\} \tag{3.1}$$

An illustration of the resulting point normals can be seen on the following page in figure 3.7. For visualisation purposes only every 400^{th} normal vector is displayed in order for the scene to still be discernible. Once the point normals are obtained these are written to a PCD file and later used in the region growing segmentation.

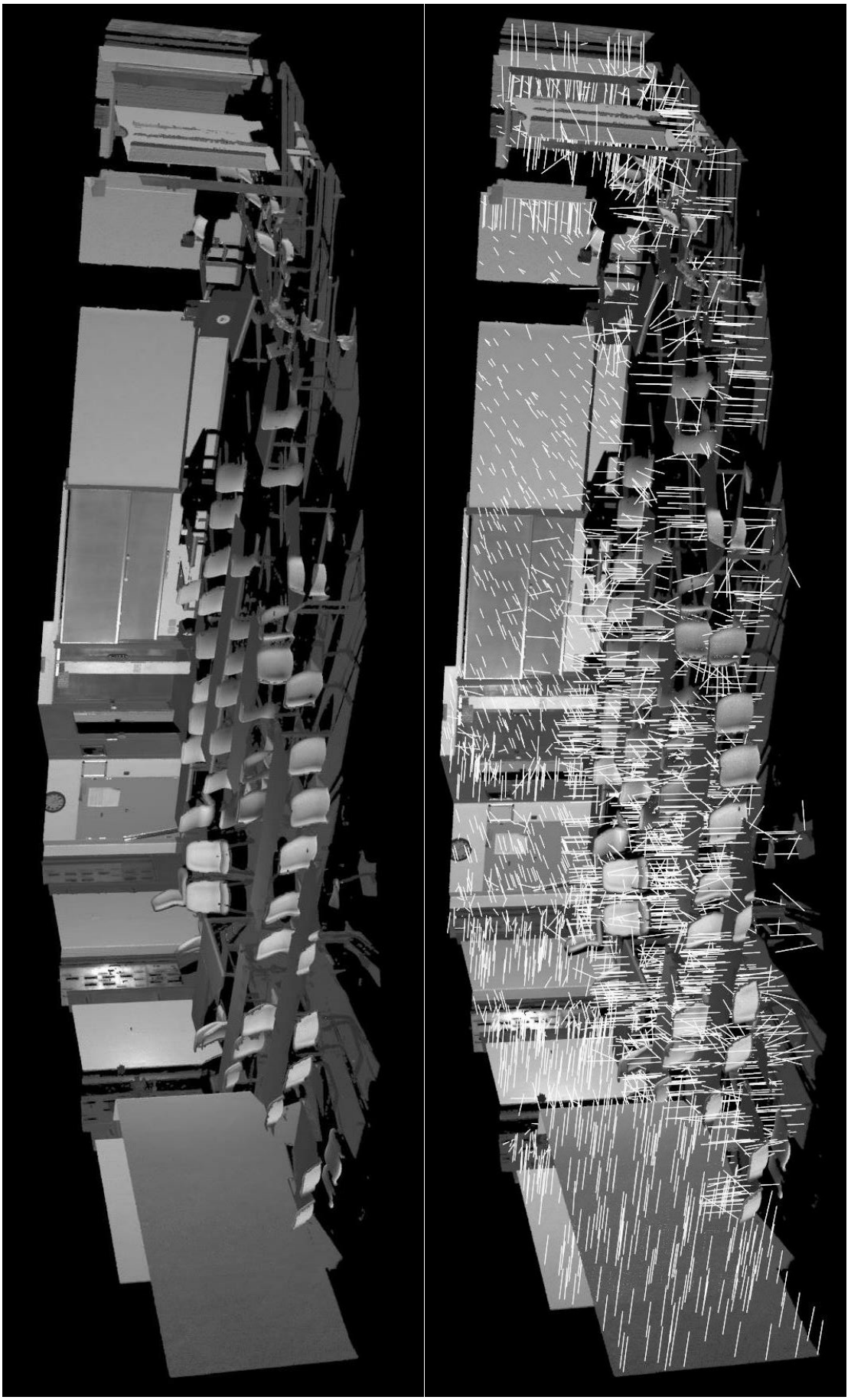


FIGURE 3.7: **Scene:** Snape 3C. **Top:** Original point cloud. **Bottom:** Point cloud illustrating every 400^{th} point normal.

3.2.2 Region Growing Segmentation

The segmentation process using the region growing algorithm from PCL is detailed below:

- Each point in the PCD is sorted by its curvature value as the region growing starts from the point that has the lowest curvature (indicating flatness). This initial point is called the seed point.
- The number of neighbouring points are selected based using the KNN algorithm and a value specified beforehand.
- The selected neighbouring points are then added to seeds.
- The angle between the selected point normal within the seeds and the initial seed point normal is determined. This angle represents the smoothness constraint and is used to determine whether the points belong to the same cluster. If the value is below the threshold then it is added to the current region.
- The deviation of the point normals for the current region is calculated. This represents the curvature value and if the newly added point results in the region exceeding the curvature threshold then the region stops growing and a new seed point for a new segment is selected.
- This procedure is repeated until there are no new points to process.

Figure 3.8 on the following page illustrates the segmented point cloud. Each segment has a different colour whilst the red points are those that do not belong to any segment. This allows a visual assessment of the segmentation in order to visually portray each segment, which has been detected, by different colours.

Once the segments have been extracted from the PCD they are stored in Random-Access Memory (RAM) instead of being written to file immediately. Additional processing on each segment must still occur and writing the segments to disk at this stage would have slowed down the computational time considerably. The next step is to discern whether the segment is horizontal or vertical which is accomplished by performing a PCA for each segment.

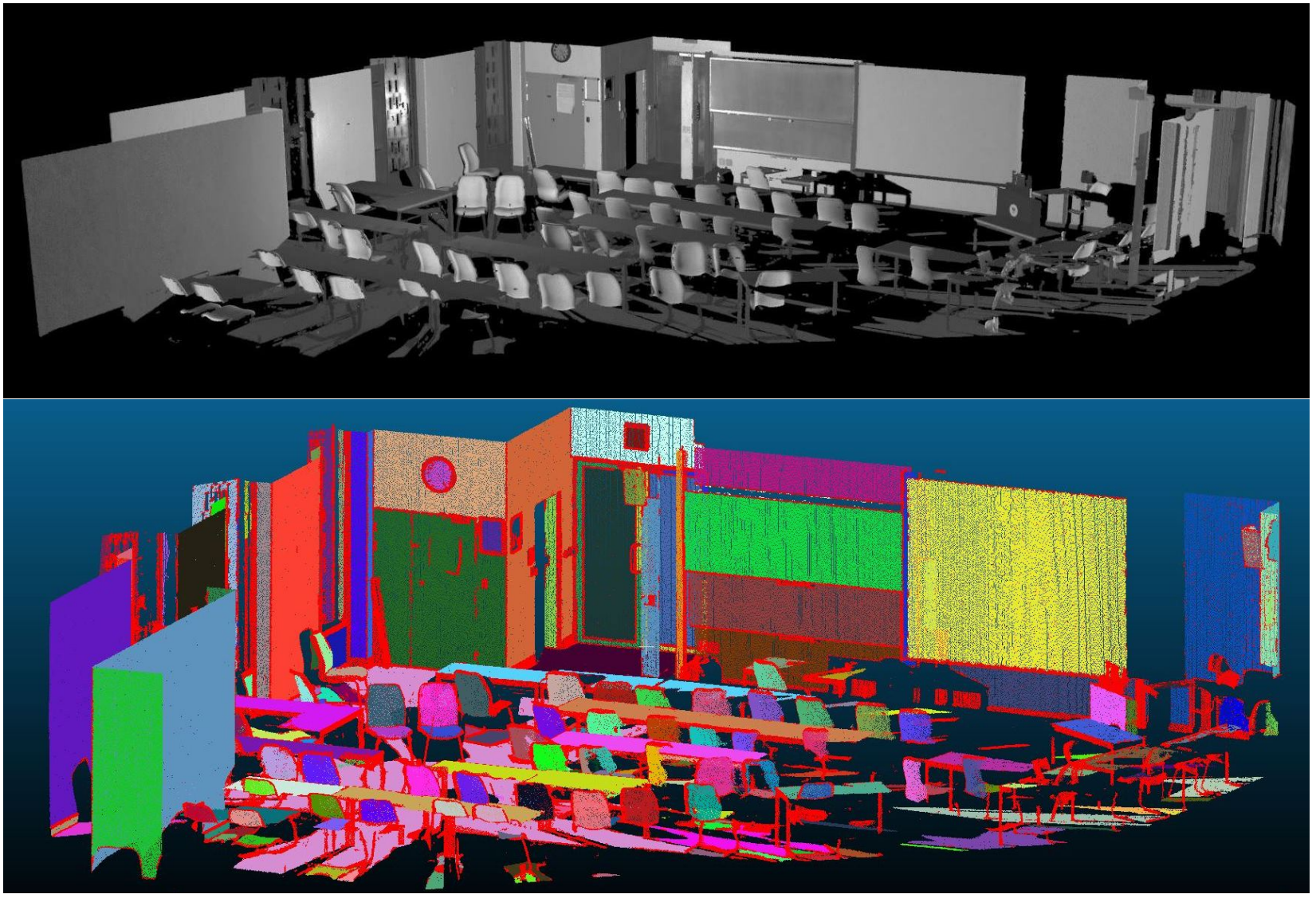


FIGURE 3.8: **Scene:** Snape 3C. **Top:** Original point cloud. **Bottom:** Point cloud illustrating each segment with a different colour. Red points do not belong to any segment.

3.2.3 PCA

A PCA is used to indicate the strongest pattern in a dataset. The first step is to calculate the eigenvector for a given dataset which will depict the direction in which the largest variance occurs. In this case the datasets are the segments which consist of a collection of 3D (X, Y, Z), coordinate points and as such the resultant eigenvectors represent the three orthogonal axes in 3D coordinate space that the points vary the most. Each eigenvector has an eigenvalue which describes how much variance is in that direction. The largest eigenvalue represents the eigenvector or direction of greatest variance for the entire dataset, this is referred to as the principal component axis (figure 3.9). The smallest eigenvalue illustrates the axis of least variance and in the case of planar segments this is the normal to the plane (figure 3.10).

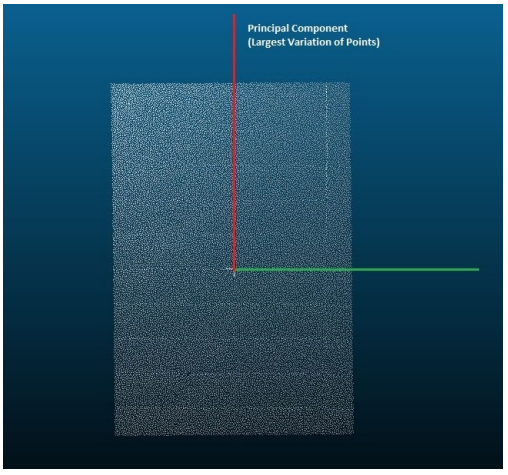


FIGURE 3.9: The red axis represents the principal component (largest variation of points)

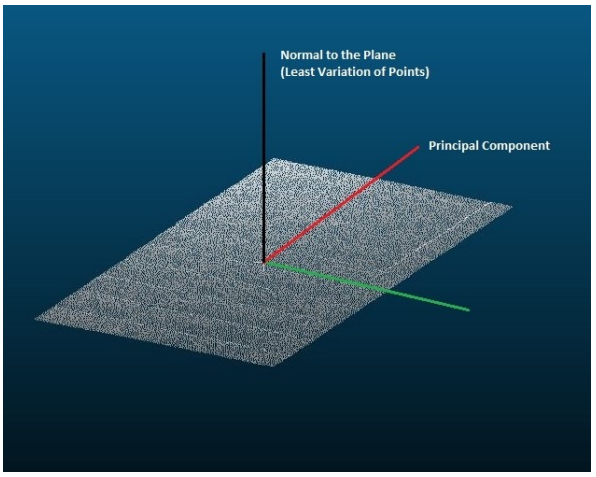


FIGURE 3.10: The black axis represents the normal to the plane (least variation of points)

The vertical is represented by the vector $\{0, 0, 1\}$ and the angle between it and the normal determines the orientation of the plane in 3D space. This angle was then calculated using equation 3.2 below where e_0 represents the minimum eigenvalue.

$$\theta = \arccos(e_0) \tag{3.2}$$

If the angle is close to 0° then it indicates a horizontal plane and if it is close to 90° it represents a vertical plane. These segments are then written to PCD files in separate folders based on whether they are horizontal or vertical. Information about each segment is written to a comma-space delimited (CSV) file for further analysis, the content of which is detailed in the next section (3.3)

3.3 Leveraging Object Recognition

The core concept of object recognition is to detect an object within a scene. Modern algorithms use feature detection and multiple descriptors but for the purposes of simplicity this was reduced to recognising objects based on their relative height from the floor (which was identified by the segment that has the lowest average height value). Once the segment has been identified as horizontal or vertical further data concerning it is written to a CSV file. This data includes heights (such as minimum, maximum and average height) and the number of points for the respective segment. The average height of the horizontal segment is used to classify whether an object is a table or chair. The height of a desk was set to lie between 72 & 80cm to allow for variation in desk types whilst the heights of chairs was set to lie between 45 & 53cm to allow for chairs with adjustable heights. Figure 3.11 below illustrates the structure of the object recognition and determining the segment orientation.

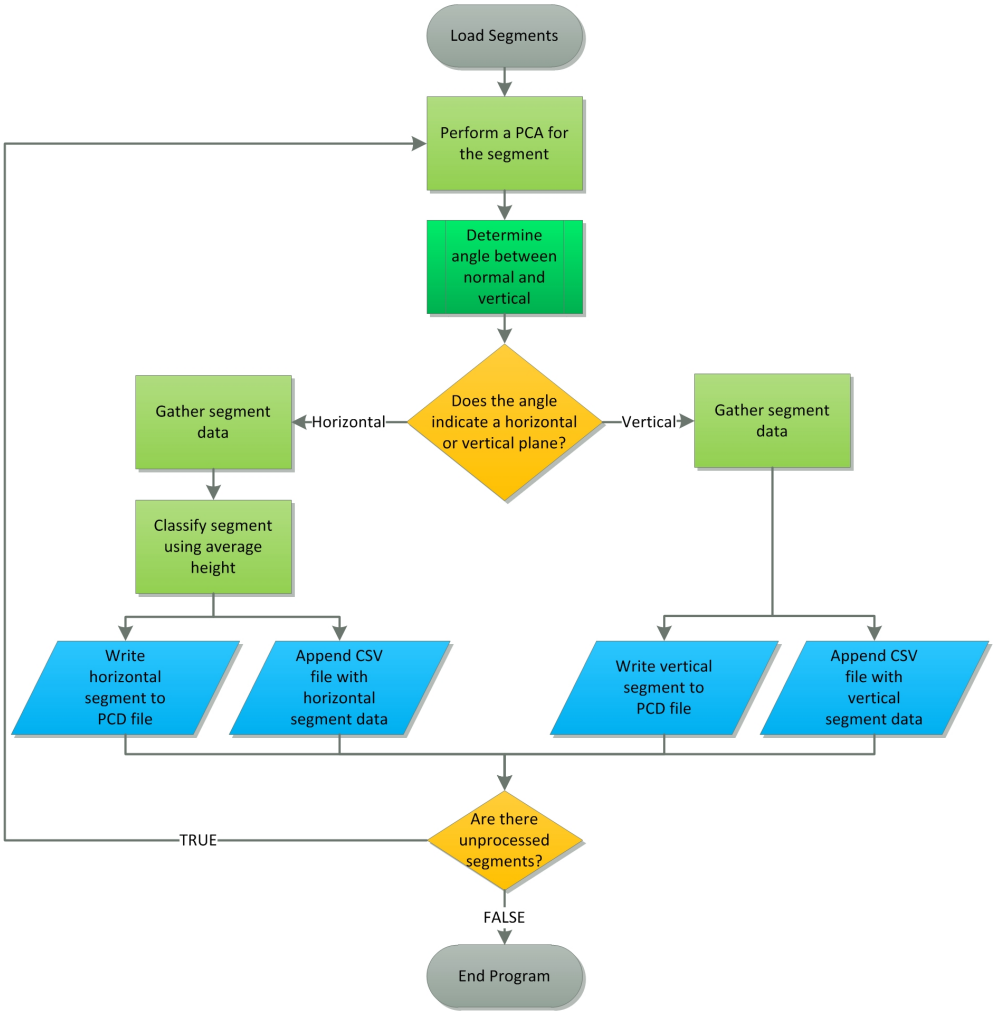


FIGURE 3.11: Workflow diagram of determining segment orientation and classification

3.4 Histogram of Heights and F-Test

Weighted histograms for the horizontal segment heights were used to illustrate the distribution of heights within a scene and to observe where spikes occur which indicated a cluster of objects. In some rooms such as the Snape 3C classroom the desks were placed very close together which resulted in strips of desks belonging to one large segment. This under-segmentation was preferred as it erred on the side of identifying an object rather than missing it. As a result each segment within the histogram of heights was weighted using the number of points in said segment.

An F-test was performed to determine whether the variance of the height of an object type in each room were equal. The hypothesis test is defined below in equation 3.3 where s_1^2 and s_2^2 represent the sample variances. A large deviation from 1 of these variances suggest that they do not belong to the same population.

$$\begin{aligned} H_0 :&\ \sigma_1^2 = \sigma_2^2 \\ H_a :&\ \sigma_1^2 \neq \sigma_2^2 \\ Test\ Statistic :&\ \frac{s_1^2}{s_2^2} \end{aligned} \tag{3.3}$$

Chapter 4

Results

This chapter focuses on the procedure followed to test the concept and steps detailed in chapter 3.1. The results obtained will be compared to the reference heights in figure 3.1 at a scene scale and at the object scale.

4.1 Instruments Used

Scenes were scanned using the Z+F IMAGER® 5010C 3D Laser scanner as it readily provided a point cloud.

4.1.1 Software Used

The following software packages were used:

- The program was written in the C++ 11 programming language
- PCL 1.7.2 with the Pre-Built binary for Visual Studio (VS) 2013 was used to process the point cloud. The library included the dependencies listed below:
  - Boost 1.57.0 provided support for multithreading and linear algebra calculations.
  - Eigen 3.2.4 was used for matrix and vector calculations.
  - The FLANN 1.8.4 library performed the nearest neighbour searches.
  - VTK 6.2.0 was used to display the point clouds.

  - QHull 2012.1 was used to determine the convex hull of a set of points.

- The following items were Integrated Development Environments (IDE's). Both were Windows 64-bit editions to allow for increased performance and they both used the VS 2013 compiler. This allowed multiple clouds to be processed simultaneously without the need to code this feature into the program.

  - VS 2013 Community Edition by Microsoft® Windows 64-bit
  - Qt 5.5.1 Open Source Cross-Platform Application Framework with Windows 64-bit VS 2013 Host

- CloudCompare 3D Point Cloud and Mesh Processing Software - Used to view the processed clouds.

- Anaconda Scientific Package for Windows 64-bit with Python 3.4 was used to write a program that could create weighted histograms:

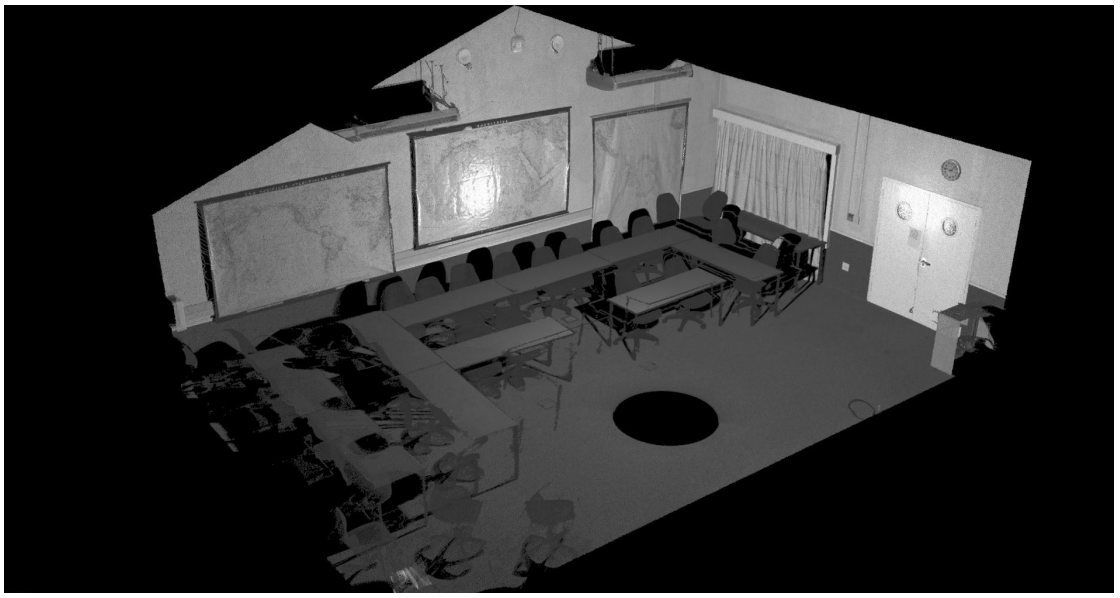
  - LiClipse 2.4.0 was the python IDE that was used.
  - Numpy 1.9.2 was used to create the weighted histogram arrays.
  - Plotly Python API 1.8.8 was used to display the graphs.

- Microsoft® Office Excel 2013 was used to perform the F-tests.

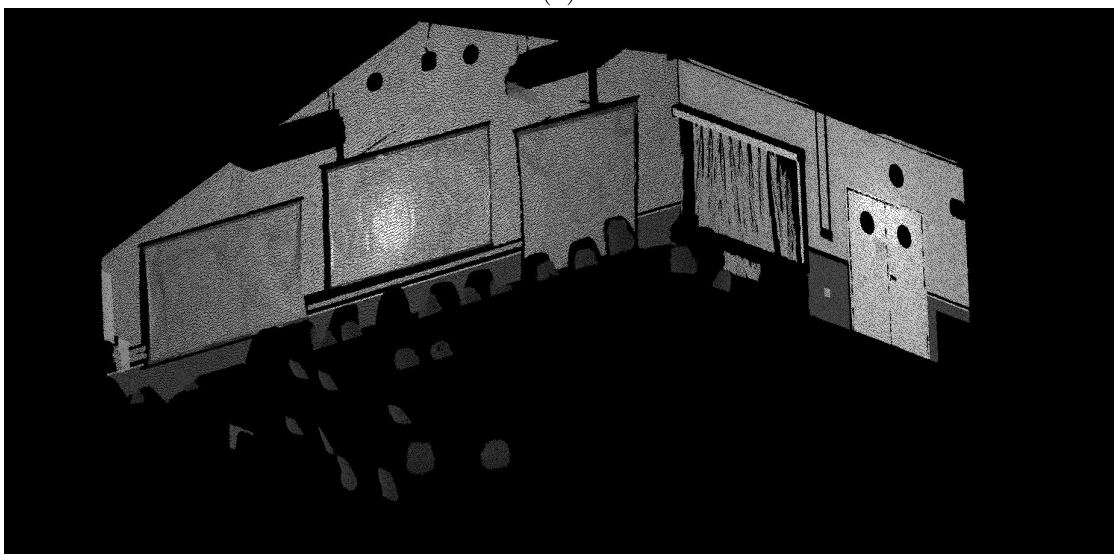
4.2 Horizontal and Vertical Planes

The angle between the normal to the plane and the vertical was used to determine whether the plane was horizontal or vertical. These segments were written to separate folders and the results of this can be seen in figures 4.1 to 4.3 on the following pages. Horizontal segments that have been recognised as desks or chairs (seats) are coloured in blue and green respectively.

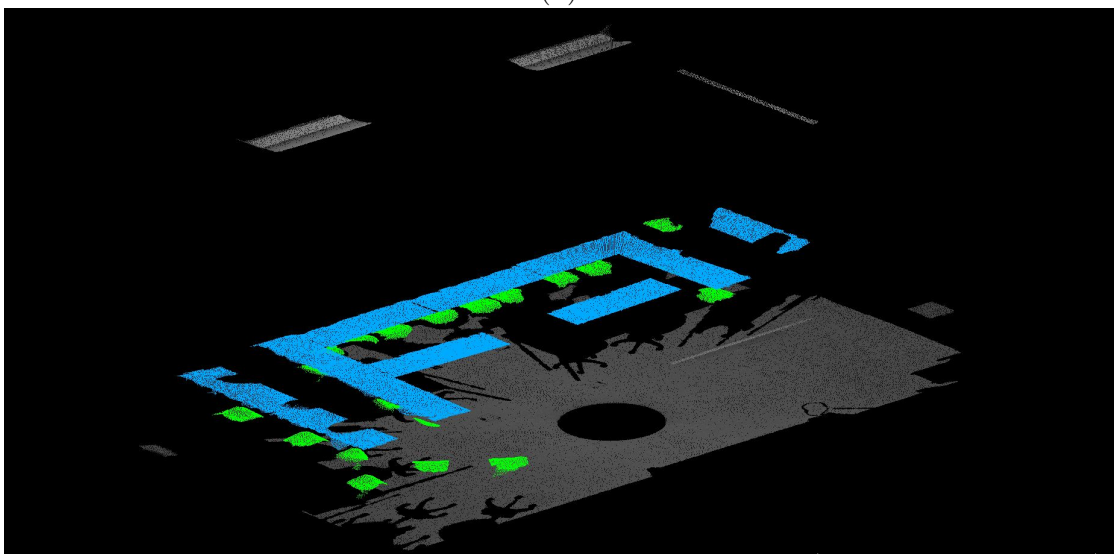
The EGS Seminar Room had the least occlusions or shadows cast which was attributed to the relatively small size of the room and the scanner height (offering a good vantage point). The Geomatics Postgraduate Computer Lab had the most occlusions for the desks (due to the scanner being placed at the same height as the cubicle dividers which obstructed the view). Snape 3C had the largest shadows cast due to the size of the room and only one scan location being used.



(a)

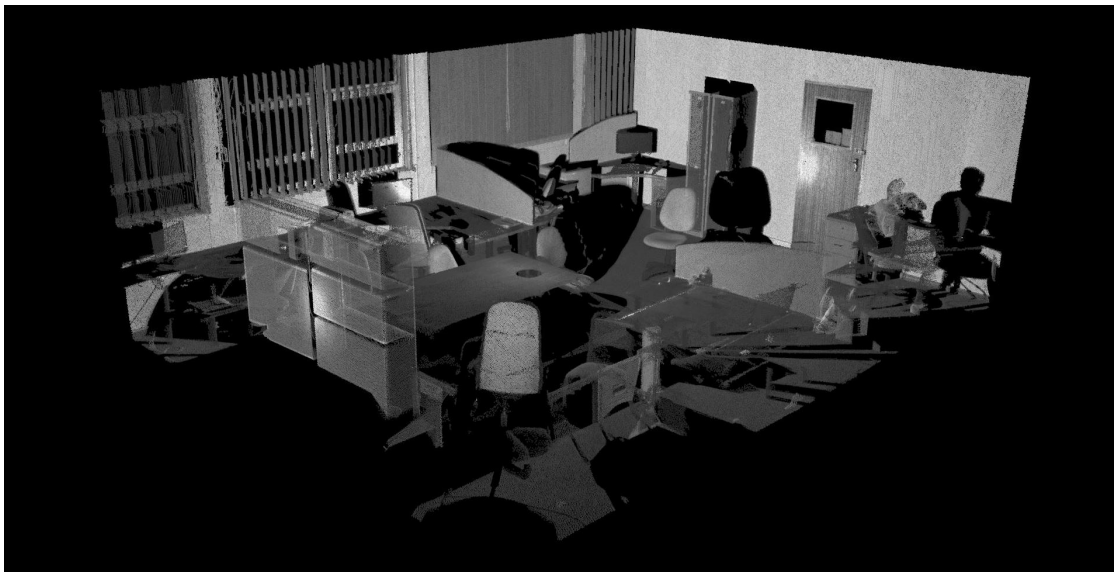


(b)

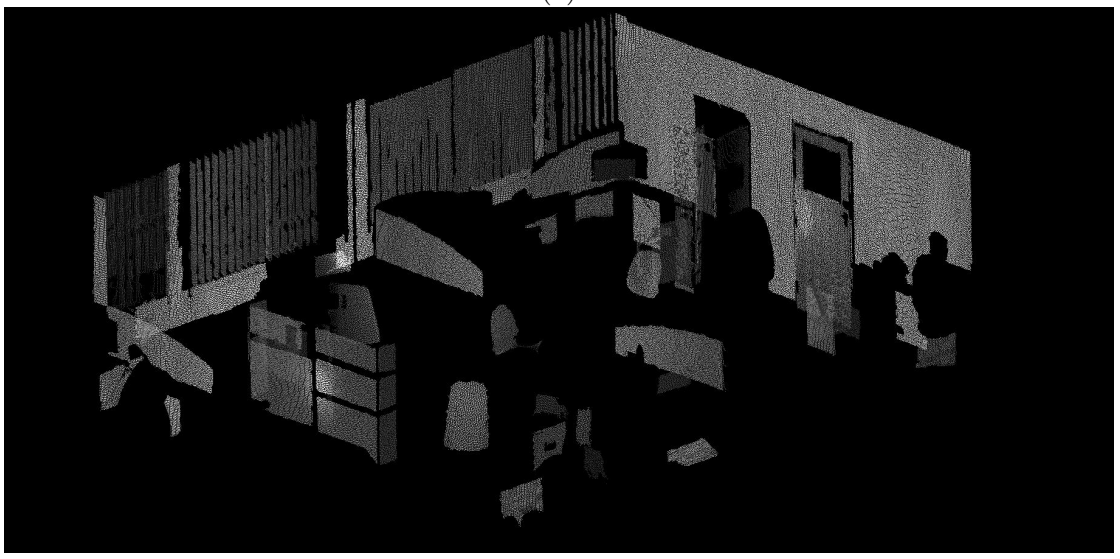


(c)

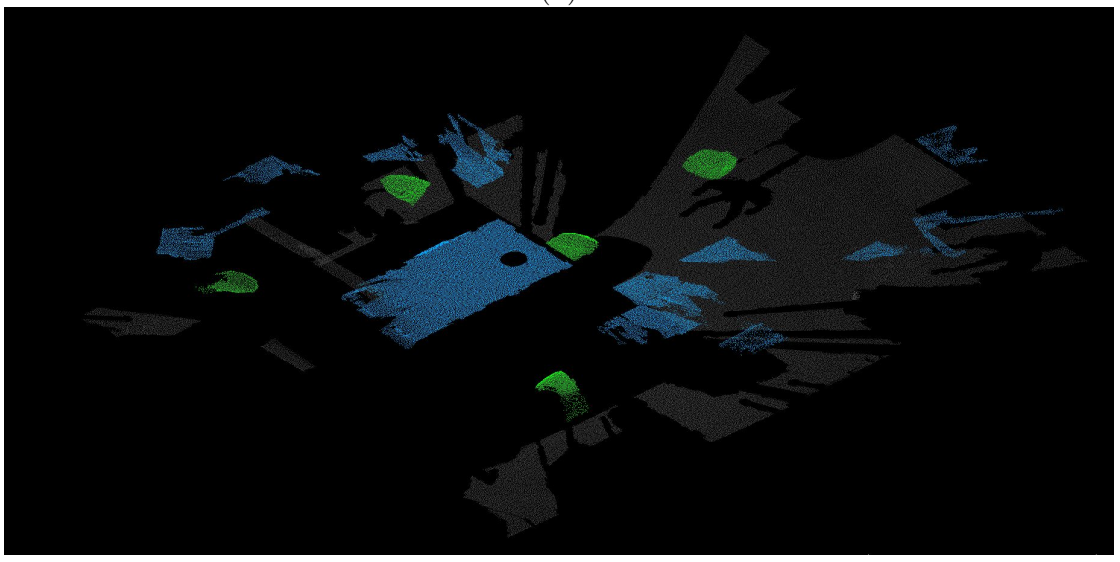
FIGURE 4.1: **(a):** EGS original point cloud. **(b):** Vertical Planes. **(c):** Horizontal Planes: Desks are coloured blue, chairs are coloured green.



(a)



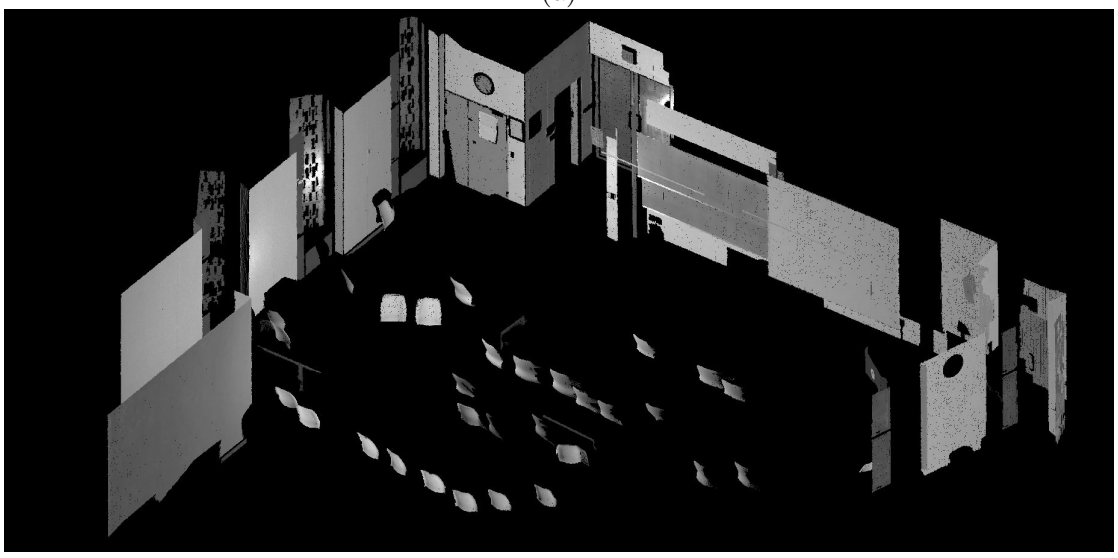
(b)



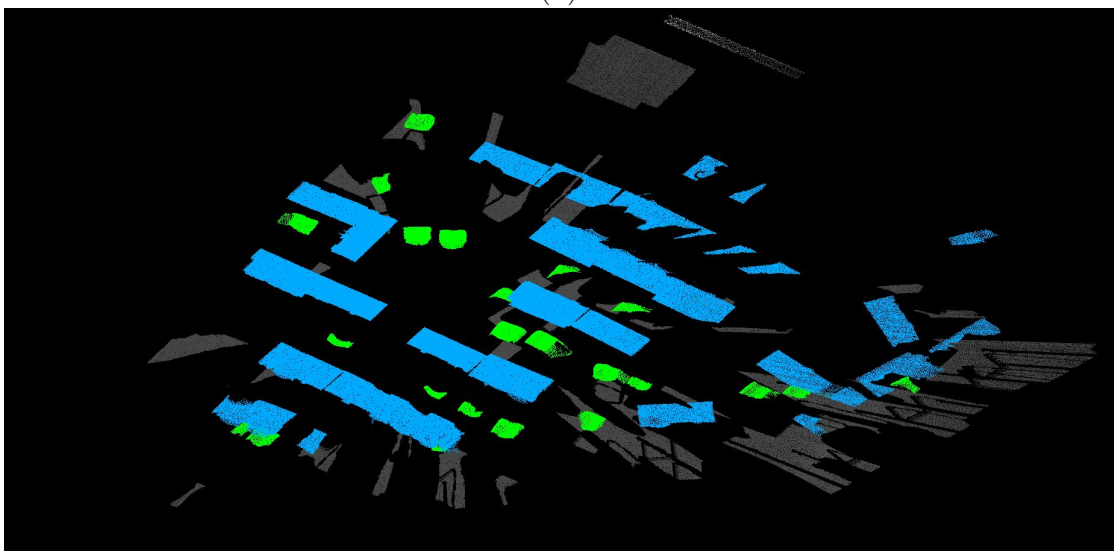
(c)

FIGURE 4.2: **(a):** Geomatics Postgraduate Lab original point cloud. **(b):** Vertical Planes. **(c):** Horizontal Planes: Desks are coloured blue, chairs are coloured green.

(a)



(b)



(c)

FIGURE 4.3: **(a):** Snape 3C original point cloud. **(b):** Vertical Planes. **(c):** Horizontal Planes: Desks are coloured blue, chairs are coloured green.

4.3 Scene Analysis

Histograms of the horizontal height segments for each scene were combined into a single graph depicting heights of segments within the scenes. Each spike in the data depicts a cluster of objects at a respective height. Objects were weighted using the number of points per segment which compensated for under-segmentation (rows of desks formed a single segment rather than separate segments). By weighting the data each spike represents the relative size of the cluster of segments at a given height. For instance the ceiling of the Geomatics Postgraduate Computer Lab, Snape 3C and the EGS Seminar Room can be identified by the spikes occurring at $2.5m$, $3.25m$ and the range between $3.5 - 3.8m$ respectively. This is verified by the low ceiling of the postgraduate lab, the large ceiling of Snape 3C (hence the very large spike at 3.25 in orange in figure 4.4) and the pointed ceiling of the EGS Seminar Room (illustrated by the pointed nature of the back wall in the top and middle of figure 3.2 and 4.1).

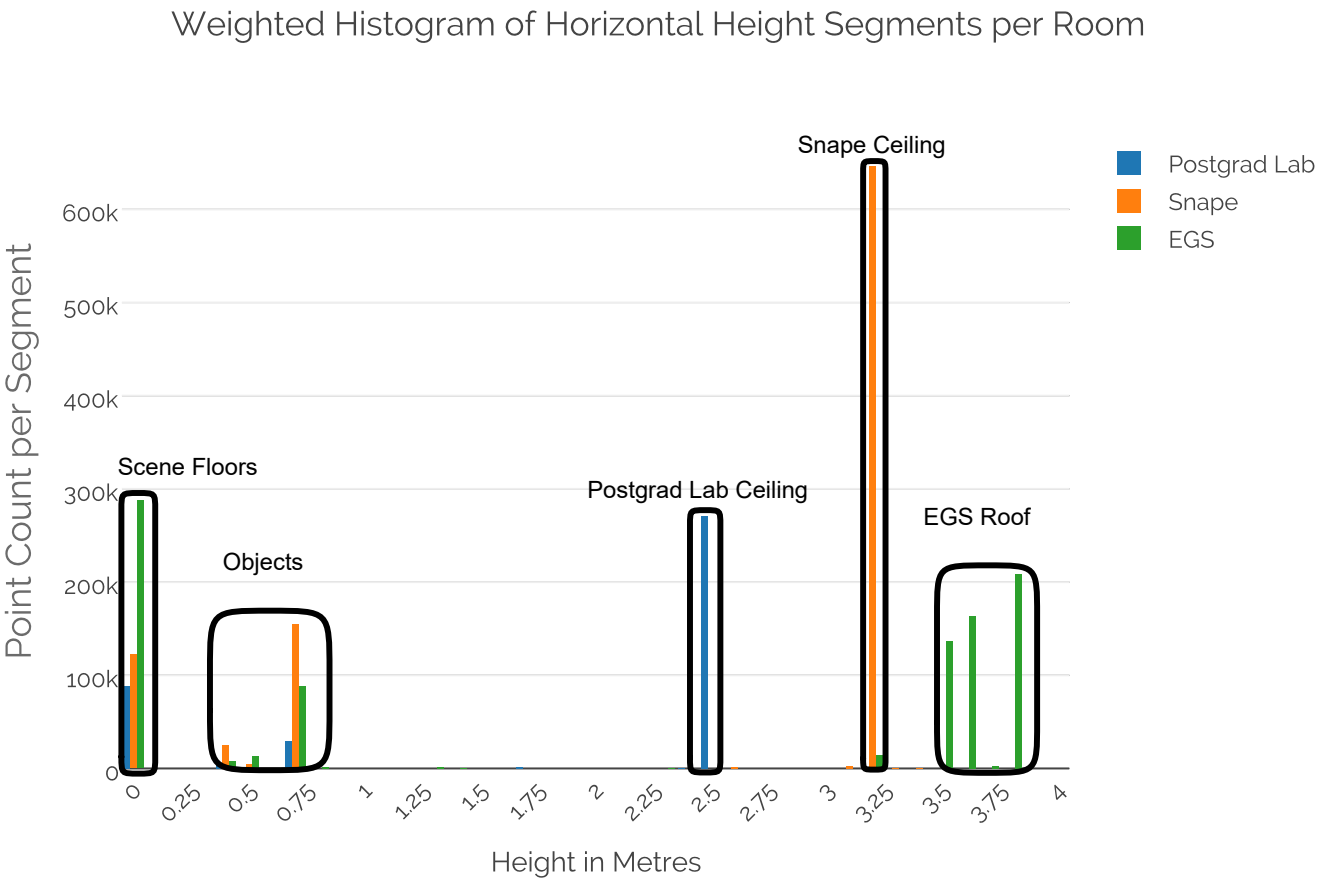


FIGURE 4.4: Histogram of height clusters per scene for the horizontal segments. These were weighted by point count per segment.

Once the spikes on the extreme ends of the graph in figure 4.4 were identified as the floors and ceiling focus was shifted to the clusters in-between. Figure 4.5 below illustrates the height range between $0.20 - 0.80m$ in greater detail. The clusters here represent objects in the form of horizontal planes at a height range that is accessible by those standing or sitting in the room. These objects are divided into two groups namely chairs and tables or desks. The spikes for desks or tables $(72 - 80cm)$ are larger than chairs $(45 - 53cm)$ due to the larger surface area they have, it does not necessarily infer that a higher number of these items were identified.

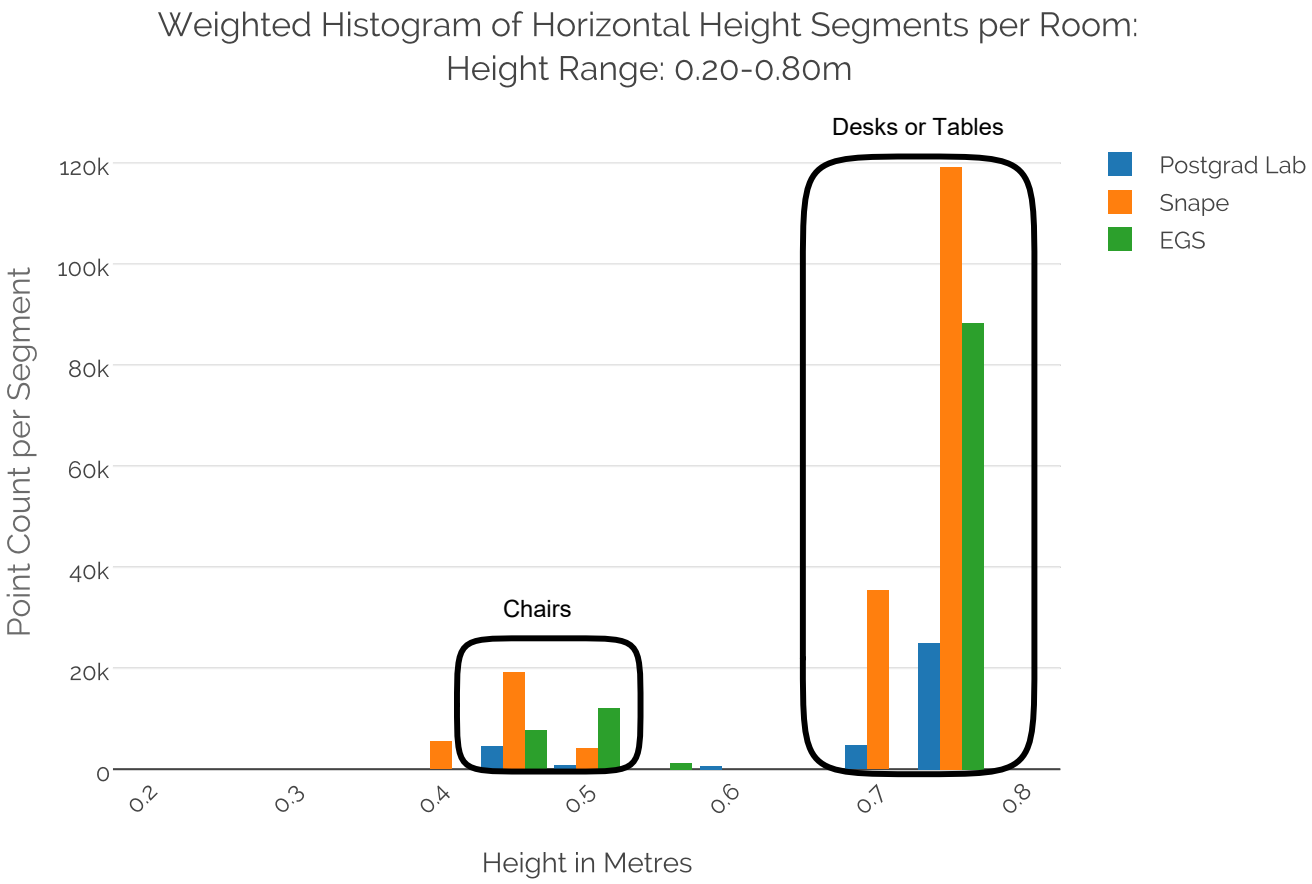


FIGURE 4.5: Histogram of horizontal segments that lie between $0.2 - 0.8m$. These are the objects within each room.

Table 4.1 illustrates the difference in height values between the reference heights listed in table 3.1 and the average height calculated from the group of classified segments for each room. The average of the values listed in table 3.1 was taken where a range of values were listed. The scale obtained for each room is depicted in figure 4.6 and compared to the average scale (represented by the bar coloured in red).

Deriving Scale from Recognised Objects					
Geomatics Postgraduate Lab					
Object	Reference Height	Average Measured Height	Scale	Difference	Ratio
Desk	74cm	75.22cm	1.016	1.22cm	1.65%
Chair	49cm	47.94cm	0.978	-1.06cm	-2.17%
		Average	0.997	0.08cm	
EGS Seminar Room					
Object	Reference Height	Average Measured Height	Scale	Difference	Ratio
Desk	74cm	76.47cm	1.033	2.47cm	3.34%
Chair	49cm	50.01cm	1.021	1.01cm	2.06%
		Average	1.027	1.74cm	
Snape 3C					
Object	Reference Height	Average Measured Height	Scale	Difference	Ratio
Desk	74cm	76.93cm	1.040	2.93cm	3.96%
Chair	49cm	48.54cm	0.991	-0.46cm	-0.94%
		Average	1.015	1.23cm	
Overall Scale: 1.013					

TABLE 4.1: Comparison of measured and reference heights for desks and chairs that were identified within each scene.

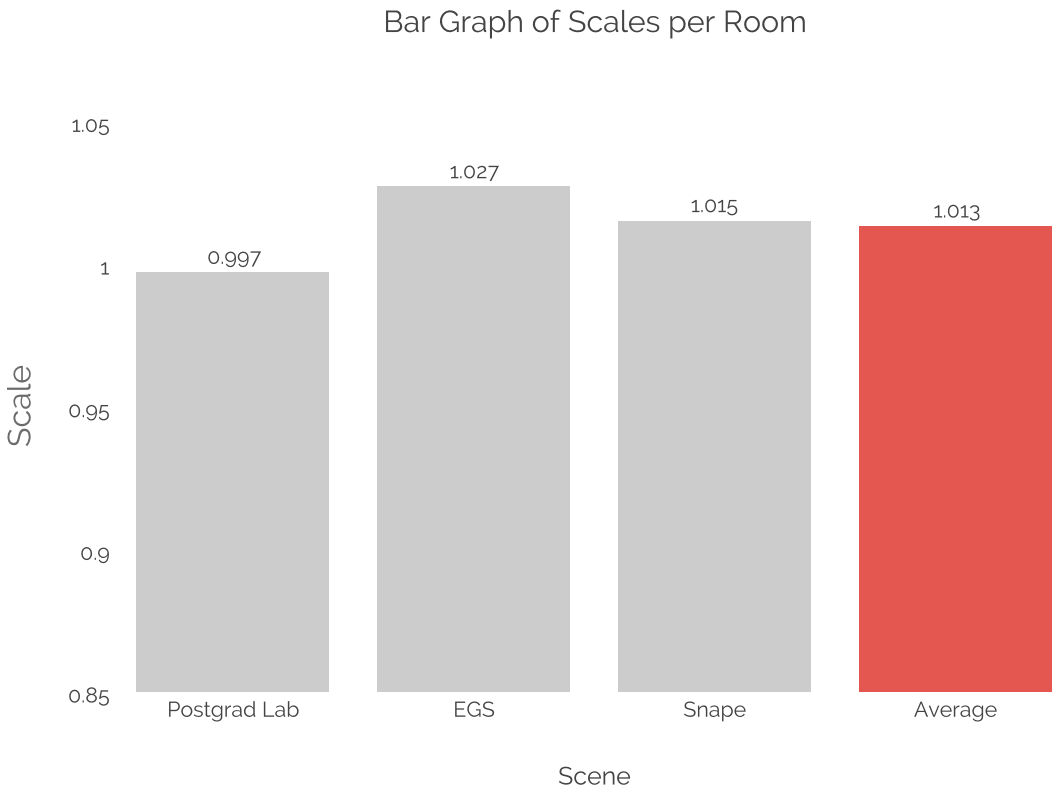


FIGURE 4.6: Bar graph of scales attained per room.

Chairs in each scene were on average $0.17cm$ lower than the reference height whilst desks were $2.21cm$ higher (calculated by taking the mean of the averages for object types in each scene illustrated in figure 4.1). This represents a ratio of -0.35% and 2.98% for chairs and desks respectively between the measured and reference heights.

Table 4.1 and figure 4.6 illustrated the discrepancy between the reference heights for chairs and desks. The scale obtained in table 4.1 was then applied to the scenes in order to bring the heights of objects closer to that of the reference heights. This was done to illustrate the difference between the heights of chairs and desks with respect to each scene rather than with respect to the reference heights. The results of this can be seen in table 4.2 and figure 4.7.

Results of Applying New Scale to Scenes					
Geomatics Postgraduate Lab					
Object	Reference Height	Average Measured Height	Scale	Difference	Ratio
Desk	74cm	74.24cm	1.003	0.24cm	0.33%
Chair	49cm	47.31cm	0.966	-1.69cm	-3.44%
		Average	0.984	-0.72cm	
EGS Seminar Room					
Object	Reference Height	Average Measured Height	Scale	Difference	Ratio
Desk	74cm	75.48cm	1.020	1.48cm	1.99%
Chair	49cm	49.36cm	1.007	0.36cm	0.73%
		Average	1.014	0.92cm	
Snape 3C					
Object	Reference Height	Average Measured Height	Scale	Difference	Ratio
Desk	74cm	75.93cm	1.026	1.93cm	2.61%
Chair	49cm	47.91cm	0.978	-1.09cm	-2.22%
		Average	1.002	0.42cm	
New Overall Scale: 1.000					

TABLE 4.2: Table showing the effect of scaling each scene such that the objects are closer to their respective reference height.

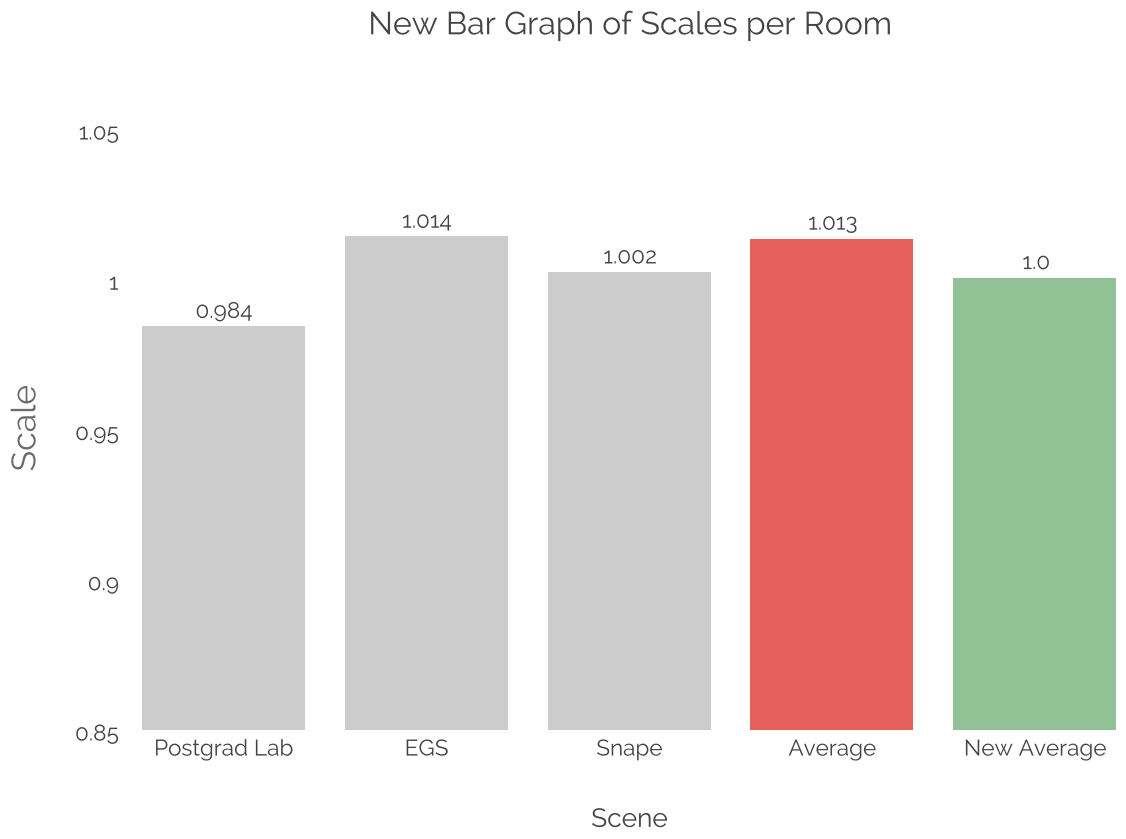


FIGURE 4.7: Bar graph illustrating the effect of applying the scale of 1.013 to each scene.

4.4 Object Analysis

The heights of objects were then normalised by dividing the height of each recognised object by $80cm$ (this is the highest a desk could be given the parameters laid out in chapter 3.3). This was done to analyse the relationship between desks and chairs rather than to the reference heights as in chapter 4.3.

The analysis of heights took place at the scene scale in the previous chapter which is why there is a larger difference in overall scene scales between tables 4.1 and 4.2 compared to tables 4.3 and 4.4 as these took place at the object scale by using normalised heights. These tables (4.3 and 4.4) along with their accompanying figures (4.8 and 4.9) can be found on the following two pages.

Deriving Scale using Normalised Heights					
Geomatics Postgraduate Lab					
Object	Reference Height	Average Measured Height	Scale	Difference	Ratio
Desk	0.92	0.94	1.016	0.015	1.65%
Chair	0.61	0.59	0.967	-0.020	-3.33%
		Average	0.992	0.003	
EGS Seminar Room					
Object	Reference Height	Average Measured Height	Scale	Difference	Ratio
Desk	0.92	0.96	1.033	0.031	3.34%
Chair	0.61	0.61	0.97	-0.002	-0.30%
		Average	1.015	0.015	
Snape 3C					
Object	Reference Height	Average Measured Height	Scale	Difference	Ratio
Desk	0.92	0.96	1.038	0.035	3.77%
Chair	0.61	0.61	0.991	-0.006	-0.94%
		Average	0.014	0.015	
Overall Scale: 1.007					

TABLE 4.3: Comparison of measured and reference heights using normalised values for identified objects within each scene.

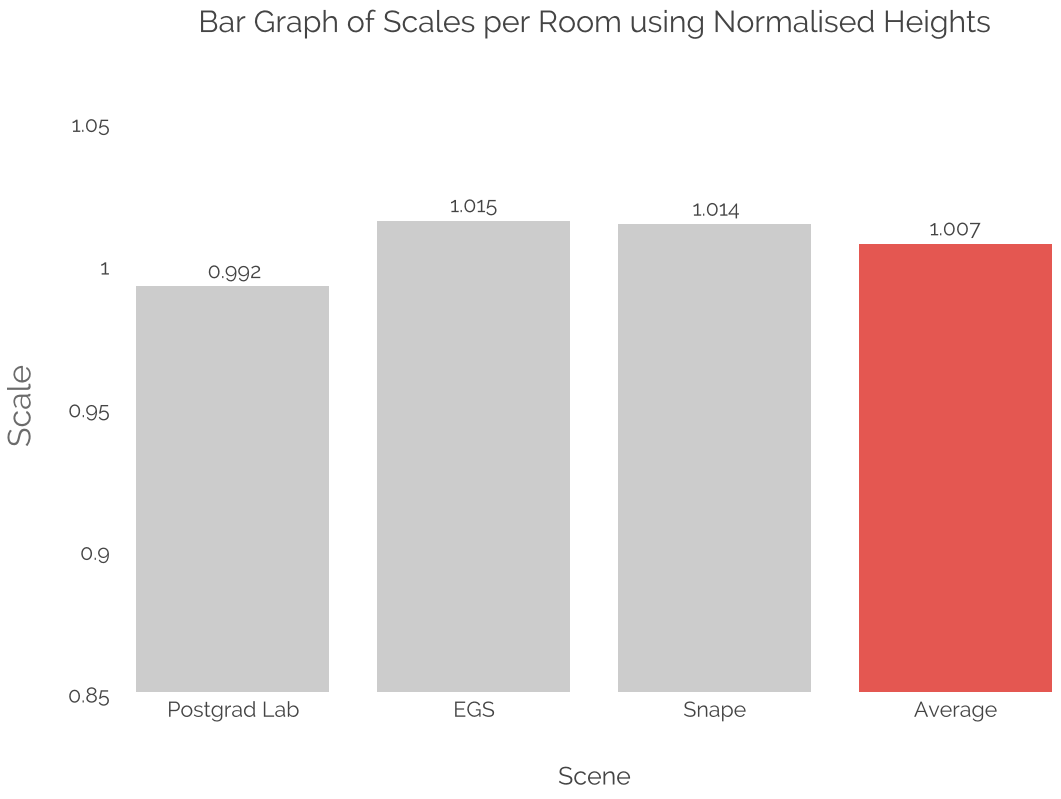


FIGURE 4.8: Bar graph illustrating the effect of applying a scale to each scene.

Results of Applying Scale Derived from Normalised Heights to Scenes					
Geomatics Postgraduate Lab					
Object	Reference Height	Average Measured Height	Scale	Difference	Ratio
Desk	0.92	0.93	1.009	0.009	0.94%
Chair	0.61	0.60	0.971	-0.017	-2.85%
		Average	0.990	-0.004	
EGS Seminar Room					
Object	Reference Height	Average Measured Height	Scale	Difference	Ratio
Desk	0.92	0.95	1.026	0.024	2.62%
Chair	0.61	0.62	1.013	0.008	1.35%
		Average	1.020	0.016	
Snape 3C					
Object	Reference Height	Average Measured Height	Scale	Difference	Ratio
Desk	0.92	0.95	1.032	0.030	3.24%
Chair	0.61	0.60	0.984	-0.010	-1.63%
		Average	1.008	0.010	
New Overall Scale: 1.006					

TABLE 4.4: Table showing the effect of scaling each scene using the scale obtained from the normalised height values.

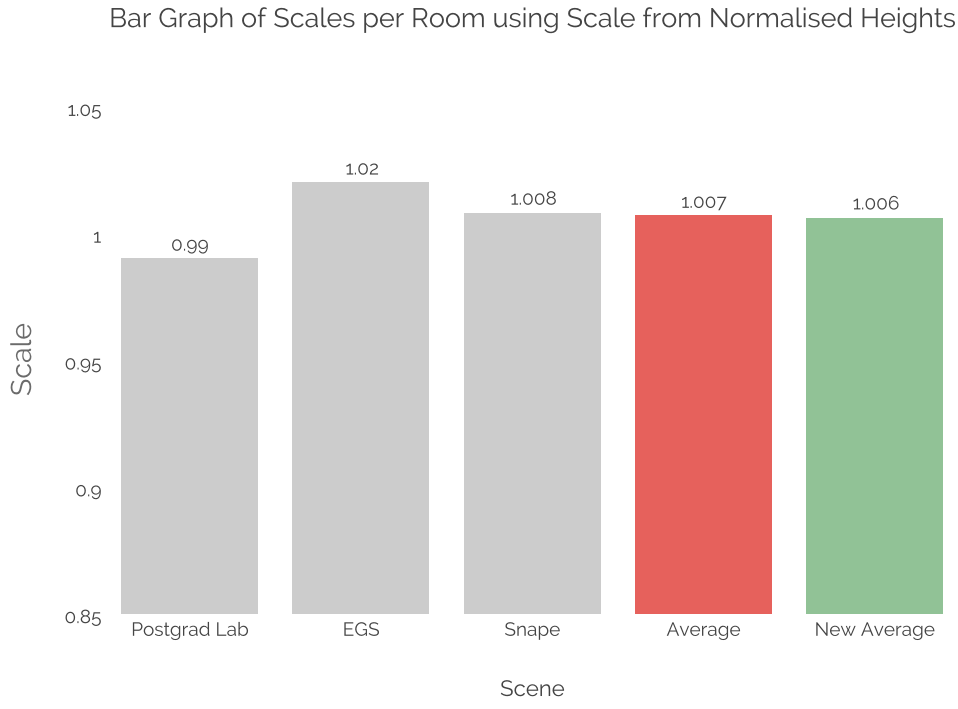


FIGURE 4.9: Bar graph illustrating the effect of applying a scale of 1.007 to each scene.

4.5 F-Test

An F-test was performed to determine if the heights for each object between scenes belonged to the same population. An example of this for chairs (using non-normalised heights) can be seen below in figure 4.5 whilst the results for all the scenes can be found in matrix form within tables 4.6 and 4.7.

F-Test Two-Sample Variances at 95% Significance Level		
Chairs	EGS	Snape
	Variable 1	Variable 2
Mean	0.500069	0.485241
Variance	0.000327	0.000271
Observations	12	19
dF	11	18
F	1.205032	
P($F <= f$) one-tail	0.350179	
F Critical one-tail	2.374155	
Verdict	PASS	

TABLE 4.5: Table showing the results of an F-Test performed in Microsoft® Office Excel 2013

F-Test Two-Sample Variances at 95% Significance Level			
Desks			
	Postgrad Lab	EGS	Snape
Postgrad Lab	-	PASS	PASS
EGS	PASS	-	PASS
Snape	PASS	PASS	-
Chair			
	Postgrad Lab	EGS	Snape
Postgrad Lab	-	PASS	PASS
EGS	PASS	-	PASS
Snape	PASS	PASS	-

TABLE 4.6: Matrix of F-Test results for non-normalised heights.

F-Test Two-Sample Variances at 95% Significance Level			
Desks			
	Postgrad Lab	EGS	Snape
Postgrad Lab	-	PASS	PASS
EGS	PASS	-	PASS
Snape	PASS	PASS	-
Chair			
	Postgrad Lab	EGS	Snape
Postgrad Lab	-	PASS	FAIL
EGS	PASS	-	PASS
Snape	FAIL	PASS	-

TABLE 4.7: Matrix of F-Test results for normalised heights.

Table 4.6 illustrates that all the scenes pass the analysis of variance test for both desks and chairs. Using normalised heights however the F-test fails when comparing the variance of heights for chairs between the Geomatics Postgraduate Lab and the Snape 3C classroom as seen in table 4.7. This can be attributed to the large difference in degrees of freedom between the two scenes (4 and 19 respectively) and the fact that using normalised heights lowers the amount of allowable variation between samples. This is because the normalised heights allow the relationship between the desks and chairs to be analysed which is at a finer scale than analysing the objects within the scene as a whole

4.6 Summary of Results

Table 4.1 illustrates the degree to which the average height of desks and chairs in a scene deviated from their respective reference height. The degree to which they differed is represented by the two columns labelled 'Difference' and 'Ratio' in table 4.1. The values within these columns and the results that follow in the upcoming paragraph were calculated using the following equations:

$$Difference = Average\ Measured\ Height\ - Reference\ Height \tag{4.1}$$

$$Ratio = \frac{Difference}{Reference\ Height} \tag{4.2}$$

The maximum ratio was 3.96% for desks which meant that the desks were 2.93cm taller than their expected (reference) height (as illustrated in figure 4.1). The smallest difference between measured and reference heights were for chairs at -0.94% which resulted in them being 0.46cm shorter than their reference height, also from figure 4.1. Both of these occurred for the newly constructed and furnished Snape 3C classroom which had the highest number of desks and chairs.

Height values were normalised in order to analyse the relationship between desks and chairs. This yielded more accurate results. The values quoted here come from tables 4.3 and 4.4. Where relevant they have been converted back to their non-normalised form by multiplying them out by 80cm as described in chapter 4.4. The largest deviation from normalised reference heights was for desks at 3.77% which resulted in a 2.8cm discrepancy. The smallest deviation using normalised values was for chairs at 0.30% which meant they were 0.16cm shorter than their normalised reference height. By analysing the relationship between chairs and desks rather than the scene as a whole a more robust scale was determined. The scale obtained through normalised heights (1.007) was applied to the scenes again and the resulting scale (1.006) only varied by a factor of 0.001 from the original (as seen in figure 4.9). Comparatively when this process was performed using non-normalised heights the difference in scale was 0.013 as seen in figure 4.7.

Figure 4.4 illustrates the difference in both ceiling height and type between the scenes. The Geomatics Postgraduate Lab and Snape 3C had flat ceilings whilst the EGS Seminar Room had an open ceiling which exposed the pointed nature of the roof. None of the rooms had the same ceiling height however which means that each room had different wall heights. Therefore walls and ceilings do not represent objects with dimensions that can serve as reliable estimates upon which to determine scale.

Chapter 5

Conclusion

Objects in a scene were recognised using the height of their horizontal plane from the ground. The average height of each object type per room was then compared to a reference height. The discrepancy between the average and reference heights was expected to lie within 10%. This meant that a deviance of $7.4cm$ for desks and $4.9cm$ for chairs from their respective reference heights was expected. The maximum measured difference using normalised values was 3.77% for desks which meant that desks were $2.8cm$ taller than their reference height (as detailed in chapter 4.6). The smallest difference between measured and reference heights were for chairs at -0.30% which resulted in them being $0.16cm$ shorter than their reference height.

The objective of this thesis was accomplished. The scale was obtained using a fully-automatic method. The proof-of-concept test yielded results below the expected 10% deviance from reference heights. The major find however was that the absolute measurement of objects yielded less accurate results compared to analysing the relationship between objects whose dimensions are based on human anatomy.

The implication of this discovery has the potential to change the approach taken when using object recognition to automatically derive scale. Rather than detecting an object in a scene whose dimensions are known and using that to derive scale for the scene the relationship between different objects whose dimensions are based on human anatomy can be used to derive scale by using typical or expected values for their dimensions.

The results obtained were from scenes on a university campus. There was variety between the scenes in terms of purpose and furniture; a computer lab (Geomatics Postgraduate Lab), a seminar room (EGS) and a modern classroom (Snape 3C). UCT does not have a policy that determines the dimensions for furniture but they do have a list of preferred vendors which has been chosen based on having met quality standards. When researching furniture policies it became clear that the dimensions concerning furniture were not controlled to the exclusion of overall size so as to fit within an office or through the door for loading purposes. This allows one to rule out the possibility of obtaining misleadingly good results during the course of this research by having used scenes on a university campus as there exists no policy which determines furniture dimensions. In order to exhaustively test the proof-of-concept scenes in a different context must be used.

Other limits of the conclusions drawn from this research is that the typical or expected values for an object whose dimensions are based on human anatomy can vary between different countries. This is because human anatomy can vary between country populations, sometimes significantly enough to warrant differently scaled furniture for instance. In order to overcome this a greater variety of scenes must be tested. This includes spaces that are not situated on a university campus such as retail spaces, offices, and places of residence.

Chapter 6

Recommendations

This section will discuss possible avenues of further development in order to offer a full-realised solution.

6.1 Object Recognition

Object recognition with multiple descriptors can be used to improve the system rather than relying on recognising objects that fall within a particular height range. This will enable more accurate identification of objects within a greater variety of scenes.

Implementing a fully-fledged object recognition system however will vary between platforms. For example desktops have more processing power and are often less restricted by internet usage and bandwidth concerns than mobile devices. Possible implementations for these two types of platforms will be discussed in the subsections that follow.

6.1.1 Development of a Smartphone Application

Due to the widespread use of smartphones across the globe emphasis should be placed on developing an application for these devices that enable them to map their environment using the hardware currently available. This can be achieved by leveraging LSD-SLAM to generate the PCD and object recognition to obtain scale on the fly. LSD-SLAM can run in real-time and modern smartphones possess the processing power to perform this locally (Caruso et al., 2015). The object recognition however can be performed on the

device or sent to a server using on-board wireless technology such as WiFi or mobile networks.

By performing the object recognition locally there needs to be a database of common objects (with a known dimension) one expects to find in any scene whether it be indoors or outdoors. This requires significant research for a number of reasons. A commonly found object may not have the same dimensions in every country. For example the average height of a chair in the United States may not be the same in Indonesia as the average height for the population may vary between each country as mentioned in chapter 5. Another concern is keeping the list of common objects short in order to minimise storage requirements for the application and to enable faster searches (as the database to search through will be smaller so the search will be faster). A fully-fledged database needs to be developed that is general enough to be used in any type of scene at any location which is efficient enough not to affect the real-time nature of the LSD-SLAM system. If all of these requirements were met then a fully-fledged mapping solution which can obtain a real-world scale could be performed directly on the device. This would be an extremely powerful tool without concerns such as cost for data transmission over mobile networks or the availability of free WiFi.

If however the object recognition could be performed by a server then the local processing power can be dedicated to the LSD-SLAM system. Frames of the video feed can be sent to a server that is capable of CBIR such as TinEye so that objects within the image can be identified. These objects can then have dimensions stored locally or on a server. The relevant scale can be sent to the device so that it can be applied to the PCD as it is being generated by the LSD-SLAM system. This allows the smartphone to access billions of images, far more than could be stored locally on the device. TinEye currently has an API that enables developers to create applications that make use of their CBIR service called MobileEngine. Using a server based approach relies on a wireless access point to be available which is not always the case. Multiple images will have to be sent which raises bandwidth concerns and possible data charges. The impact of using WiFi on battery life is also a factor that needs to be taken into account.

Other areas not related to determining scale need to be assessed such as the memory requirements of such a system as massive amounts of data need to be captured in terms of video frames or multiple images and then converted to a point cloud representation. This

affects both the RAM and local storage specifications. These problems have however been tackled by Google's Project Tango where a tablet variant is on offer which makes use of a 4 megapixel RGB-Infrared sensor that has access to 4Gb of RAM and a 128Gb SSD for rapid local read and write storage.

6.1.2 Web-based Applications

There are billions of images on the internet which document places on Earth, many of which are well known. CBIR systems can sift through image databases such as flickr to find images of a particular scene such as a monument. SfM can then be used to combine images taken at different times with a variety of cameras. Object recognition that makes use of services such as TinEye can then identify objects within the images in order to derive a scale for the PCD. This system can be built into a web-based application that users can interact with through their web browser with all the processing being performed remotely on a server. The server could also have country-specific dimensions for common objects that are based on human anatomy for both indoor and outdoor spaces. Users can also have the option of uploading their own images with the aim of recreating a 3D environment where they are the subject of the scene.

Both the private and government sectors can use this technology for a variety of tasks. Shopping malls can be digitally reconstructed and scaled using commonly found objects such as benches so that customers can access a 3D map and path finding system to find points of interest. Due to the size of a structure such as a mall this task would be more easily performed on a desktop once all the imagery had been uploaded to it. This concept can be extended to other areas, for example theme parks or museums such as the Louvre in Paris. Law enforcement can use this technology to derive real-world measurements such as heights of people from surveillance or public footage.

Bibliography

Anon. (2011), Vision Recognition, Technical report, Brigham Young University, Department of Electrical and Computer Engineering, Provo, Utah, United States. Accesssed October 14th, 2015.
URL: *http://rwbclasses.groups.et.byu.net/lib/exe/fetch.php?media=campus_challenge:vr_documentation.pdf*

Ayache, N. (1991), *Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception*, MIT Press.

Bay, H., Tuytelaars, T. and Van Gool, L. (2006), Surf: Speeded up robust features, *in* 'Computer visionECCV 2006', Springer, pp. 404–417. Accessed 14 October 2015.
URL: *http://link.springer.com/chapter/10.1007/11744023_32*

Beardsley, P., Reid, I. D., Zisserman, A., Murray, D. W. and others (1995), Active visual navigation using non-metric structure, *in* 'Computer Vision, 1995. Proceedings., Fifth International Conference on', IEEE, pp. 58–64. Accessed 15 October 2015.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=466806*

Bormann, R. and Hampp, J. (2014), 'Innovative image processing for service robots'. Accessed 14 October 2015.
URL: *http://www.nanowerk.com/news2/robotics/newsid=34797.php*

Caruso, D., Engel, J. and Cremers, D. (2015), Large-Scale Direct SLAM for Omnidirectional Cameras, *in* 'International Conference on Intelligent Robots and Systems (IROS)'.

David Sachs (2010), 'Sensor Fusion on Android Devices: A Revolution in Motion Processing'. Accessed 16 October 2015.

URL: *https://www.youtube.com/watch?v=C7JQ7Rpwn2k&feature=youtu.be&t=23m21s*

Davidson-Pilon, C. (2012), 'machine learning - What is "feature space"? - Cross Validated'. Accessed 6 October 2015.
URL: *http://stats.stackexchange.com/questions/46425/what-is-feature-space*

Davison, A. J. (2003), Real-time simultaneous localisation and mapping with a single camera, *in* 'Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on', IEEE, pp. 1403–1410. Accessed 15 October 2015.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1238654*

DevZone, O. (2011), 'OpenCV Meeting notes for 2011 year'. Accessed 14 October 2015.
URL: *http://code.opencv.org/projects/opencv/wiki/2011*

Engel, J., Schps, T. and Cremers, D. (2014), LSD-SLAM: Large-scale direct monocular SLAM, *in* 'Computer VisionECCV 2014', Springer, pp. 834–849. Accessed 15 October 2015.
URL: *http://link.springer.com/chapter/10.1007/978-3-319-10605-2_54*

Fergus, R. (2012), 'Lecture 6: Multi-view Stereo & Structure from Motion'. Accessed 19 September 2015.
URL: *http://cs.nyu.edu/~fergus/teaching/vision_2012/6_Multiview_SfM.pdf*

Fitzgibbon, A. W. and Zisserman, A. (1998), Automatic camera recovery for closed or open image sequences, *in* 'Computer VisionECCV'98', Springer, pp. 311–326. Accessed 15 October 2015.
URL: *http://link.springer.com/content/pdf/10.1007/BFb0055675.pdf*

Fleet, D., Pajdla, T., Schiele, B. and Tuytelaars, T. (2014), *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings*, Springer.

Goldstein, P. (2014), 'Report: Global smartphone penetration to jump 25% in 2014, led by Asia-Pacific'. Accessed 16 October 2015.
URL: *http://www.fiercewireless.com/story/report-global-smartphone-penetration-jump-25-2014-led-asia-pacific/2014-06-11*

Griggs, J. M. (2001), 'Typical Furniture Measurements'. Accessed 22 October 2015.
URL: *http://www.fas.harvard.edu/~loebinfo/loebinfo/Proportions/furniture.html*

Harris, C. (1992), Geometry from visual motion, *in* A. Blake, ed., 'Active Vision', MIT Press, pp. 263–284.

Insider, B. (2015), The Drones Report: Market forecasts, regulatory barriers, top vendors, and leading commercial applications, Technical report. Accessed 26 October 2015.
URL: *http://www.businessinsider.com/uav-or-commercial-drone-market-forecast-2015-2*

Klein, G. and Murray, D. (2007), Parallel tracking and mapping for small AR workspaces, *in* 'Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on', IEEE, pp. 225–234. Accessed 15 October 2015.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4538852*

Lai, K., Bo, L., Ren, X. and Fox, D. (2013), RGB-D object recognition: Features, algorithms, and a large scale benchmark, *in* 'Consumer Depth Cameras for Computer Vision', Springer, pp. 167–192. Accessed 14 October 2015.
URL: *http://link.springer.com/chapter/10.1007/978-1-4471-4640-7_9*

Lefler, R. K. (2004), 'Choosing the Right Ergonomic Office Chair'. Accessed 25 October 2015.
URL: *http://www.spine-health.com/wellness/ergonomics/office-chair-choosing-right-ergonomic-office-chair*

Lepetit, V., Vacchetti, L., Thalmann, D. and Fua, P. (2003), Fully automated and stable registration for augmented reality applications, *in* 'Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality', IEEE Computer Society, p. 93. Accessed 15 October 2015.
URL: *http://dl.acm.org/citation.cfm?id=946792*

McCann, S. (2015), 3d Reconstruction from Multiple Images, Technical report, Stanford University. Accessed 13 October 2015.
URL: *http://cs.nyu.edu/~fergus/teaching/vision_2012/6_Multiview_SfM.pdf*

Mirzaei, F. and Roumeliotis, S. (2008), 'A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation', *IEEE Transactions on Robotics* **24**(5), 1143–1156.

Molinos, E. J., Llamazares, n., Hernndez, N., Arroyo, R., Cela, A., Yebes, J. J., Ocaa, M. and Bergasa, L. M. (2014), Perception and Navigation in Unknown Environments: The DARPA Robotics Challenge, *in* M. A. Armada, A. Sanfeliu and M. Ferre, eds, 'ROBOT2013: First Iberian Robotics Conference', number 253 *in* 'Advances in Intelligent Systems and Computing', Springer International Publishing, pp. 321–329. Accessed 15 October 2015.
URL: *http://link.springer.com/chapter/10.1007/978-3-319-03653-3_24*

Mostofi, N., Elhabiby, M. and El-Sheimy, N. (2014), Indoor localization and mapping using camera and inertial measurement unit (IMU), *in* 'Position, Location and Navigation Symposium - PLANS 2014, 2014 IEEE/ION', pp. 1329–1335.

Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F. and Sayd, P. (2006), Real Time Localization and 3d Reconstruction, *in* '2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition', Vol. 1, pp. 363–370.

Mur-Artal, R., Montiel, J. M. M. and Tardos, J. D. (2015), 'ORB-SLAM: a Versatile and Accurate Monocular SLAM System', *IEEE Transactions on Robotics* **31**(5), 1147–1163. Accessed 15 October 2015.
URL: *http://arxiv.org/abs/1502.00956*

Ntzi, G., Weiss, S., Scaramuzza, D. and Siegwart, R. (2011), 'Fusion of IMU and vision for absolute scale estimation in monocular SLAM', *Journal of intelligent & robotic systems* **61**(1-4), 287–299. Accessed 11 October 2015.
URL: *http://link.springer.com/article/10.1007/s10846-010-9490-z*

Pei, L., Liu, J., Guinness, R., Chen, Y., Kuusniemi, H. and Chen, R. (2012), 'Using LS-SVM Based Motion Recognition for Smartphone Indoor Wireless Positioning', *Sensors* **12**(5), 6155–6175. Accessed 16 October 2015.
URL: *http://www.mdpi.com/1424-8220/12/5/6155*

Penelope (2013), 'image processing - Object detection versus object recognition - Signal Processing Stack Exchange'. Accessed 14 October 2015.

URL: *http://dsp.stackexchange.com/questions/12940/object-detection-versus-object-recognition*

Pillai, S. and Leonard, J. (2015), 'Monocular SLAM Supported Object Recognition', *arXiv preprint arXiv:1506.01732* . Accessed 15 October 2015.
URL: *http://arxiv.org/abs/1506.01732*

Rabbani, T., van den Heuvel, F. and Vosselmann, G. (2006), 'Segmentation of point clouds using smoothness constraint', *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* **36**(5), 248–253. Accessed 26 September 2015.
URL: *http://www.isprs.org/proceedings/XXXVI/part5/paper/RABB_639.pdf*

Rashidi, A., Brilakis, I. and Vela, P. (2014), 'Generating Absolute-Scale Point Cloud Data of Built Infrastructure Scenes Using a Monocular Camera Setting', *Journal of Computing in Civil Engineering* **0**(0), 04014089. Accessed 15 October 2015.
URL: *http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000414*

Richter, F. (2012), 'Infographic: 4.4 Billion Camera Phones...'. Accessed 26 October 2015.
URL: *http://www.statista.com/chart/653/prevalence-of-selected-features-in-the-global-installed-base-of-mobile-phones/*

Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M. and Beetz, M. (2008), 'Towards 3d Point cloud based object maps for household environments', *Robotics and Autonomous Systems* **56**(11), 927–941. Accessed 28 September 2015.
URL: *http://www.sciencedirect.com/science/article/pii/S0921889008001140*

Sadiku, M. N. O. and Ali, W. H. (2015), *Signals and Systems: A Primer with MATLAB*, CRC Press.

Scaramuzza, D., Fraundorfer, F., Pollefeys, M. and Siegwart, R. (2009), Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints, *in* 'Computer Vision, 2009 IEEE 12th International Conference on', IEEE, pp. 1413–1419. Accessed 10 October 2015.
URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5459294*

Seitz, S. M., Curless, B., Diebel, J., Scharstein, D. and Szeliski, R. (2006), A comparison and evaluation of multi-view stereo reconstruction algorithms, *in* 'Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on', Vol. 1, IEEE, pp. 519–528. Accessed 9 October 2015.

URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1640800*

Strasdat, H., Montiel, J. M. M. and Davison, A. J. (2010), Scale Drift-Aware Large Scale Monocular SLAM. Accessed 15 October 2015.

URL: *http://webdiis.unizar.es/~josemari/strasdat_etal_rss2010.pdf*

Tsai, R. (1987), 'A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf TV cameras and lenses', *IEEE Journal of Robotics and Automation* **3**(4), 323–344.

Vedula, S., Rander, P., Collins, R. and Kanade, T. (2005), 'Three-dimensional scene flow', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **27**(3), 475–480. Accessed 9 October 2015.

URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1388274*

Wakefield, J. and News, B. B. C. (2014), 'One wi-fi hotspot for every 150 people, says study'. Accessed 16 October 2015.

URL: *http://www.bbc.com/news/technology-29726632*

Weiss, S., Newcombe, R. and Beall, C. (2014), 'Visual SLAM Tutorial | at CVPR 2014'. Accessed 15 October 2015.

URL: *http://frc.ri.cmu.edu/~kaess/vslam_cvpr14/*

Whitaker, R. T., Crampton, C., Breen, D. E., Tuceryan, M. and Rose, E. (1995), Object calibration for augmented reality, *in* 'Computer Graphics Forum', Vol. 14, Wiley Online Library, pp. 15–27. Accessed 15 October 2015.

URL: *http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8659.1995.cgf143_0015.x/abstract*

Ye, S. (2015), 'Smartphone Futurology: The science behind your next phone's processor and memory'. Accessed 16 October 2015.

URL: *http://www.androidcentral.com/smartphone-futurology-3-chips*

Yeh, T., Tollmar, K. and Darrell, T. (2004), Searching the web with mobile images for location recognition, *in* 'Computer Vision and Pattern Recognition, 2004. CVPR

2004. Proceedings of the 2004 IEEE Computer Society Conference on', Vol. 2, IEEE, pp. II–76. Accessed 14 October 2015.

URL: *http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1315147*